2016.07.31版

第3部：NGS解析（中～上級）
～ Linux環境でのデータ解析：JavaやRの利用法～

東京大学・大学院農学生命科学研究科
アグリバイオインフォマティクス教育研究プログラム
門田幸二（かどた　こうじ）
kadota@iu.a.u-tokyo.ac.jp
http://www.iu.a.u-tokyo.ac.jp/~kadota/

Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

まずはFaQCs実行

①乳酸菌NGS連載第5回のサイト。②のあたりまでページ下部（概ね4ページ分）に移動。③のあたりにコピペ用コマンドがあります

(Rで)塩基配列解析

～NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モデ

(last mo

What's

- このウ
リーソ
法(W

- 書籍 | 日本乳酸菌学会誌 | について (last modified 2016/05/12) NEW
- 書籍 | 日本乳酸菌学会誌 | 第1回イントロダクション (last modified 2015/09/11)
- 書籍 | 日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境へ (last modified 2
- 書籍 | 日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ取得まで (last modi
- 書籍 | 日本乳酸菌学会誌 | 第4回クオリティコントロールとプログラムのインストール
- 書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC modified 2
- 書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ (last modified 2016 /23)
- 書籍 | 日
- イントロ |

①

書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC

日本乳酸菌学会誌の第5回分です。Linuxコマンドのリンク先はほぼ日経BP社様です。

- 第5回分PDF
- ウェブ資料PDF(2

Linuxコマンド

- bzip2 (bzip2圧縮
- cd (ディレクトリを
- echo (文字列を
- export (変数を
- file (ファイルタイ

②

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.3)を実行。
連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
fastqc2 -q SRR616268sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

連載第4回[W9-5]の手順通りに行ったヒトは、fastqc2がFastQC ver. 0.11.3の実行プログラムになっているはずです。実行結果ファイルSRR616268sub_1_fastqc.html(forward側)では、(古いバージョンFastQC ver. 0.10.1の結果ですが)連載第4回W8-6に示すように、Overrepresented sequencesという項目中に「TruSeq Adapter, Index2と3」がリード中に含まれていることがわかります。また、SRR616268sub_2_fastqc.html (reverse側)では、「Illumina Single End PCR Primer 1」がリード中に含まれていることがわかります。

- Skewer: Jiang et al., BMC Bioinformatics, 2014
- FaQCs: Lo and Chain, BMC Bioinformatics, 2

③

- 図1。第4回の[W17-3]と基本的に同じ。[W1-1]

```
cd ~/Documents/snp017156
```

まずはFaQCs実行

①ここです。②作業ディレクトリを変更。③は無視でよい（第4回最後の状態から不必要なものを削除しているだけ）。②をコピペ後に、④をまずはコピペ実行。約20分

スライド18まで説明

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.3)を実行。
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
fastqc2 -q SRR616268sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

連載第4回[W9-5]の手順通りに行ったヒトは、fastqc2がFastQC ver. 0.11.3の実行プログラムになっているはずです。実行結果ファイルSRR616268sub_1_fastqc.html(forward側)では、(古いバージョンFastQC ver. 0.10.1の結果ですが)連載第4回W8-6に示すように、Overrepresented sequencesという項目中に「TruSeq Adapter, Index2と3」がリード中に含まれていることがわかります。また、SRR616268sub_2_fastqc.html (reverse側)では、「Illumina Single End PCR Primer 1」がリード中に含まれていることがわかります。

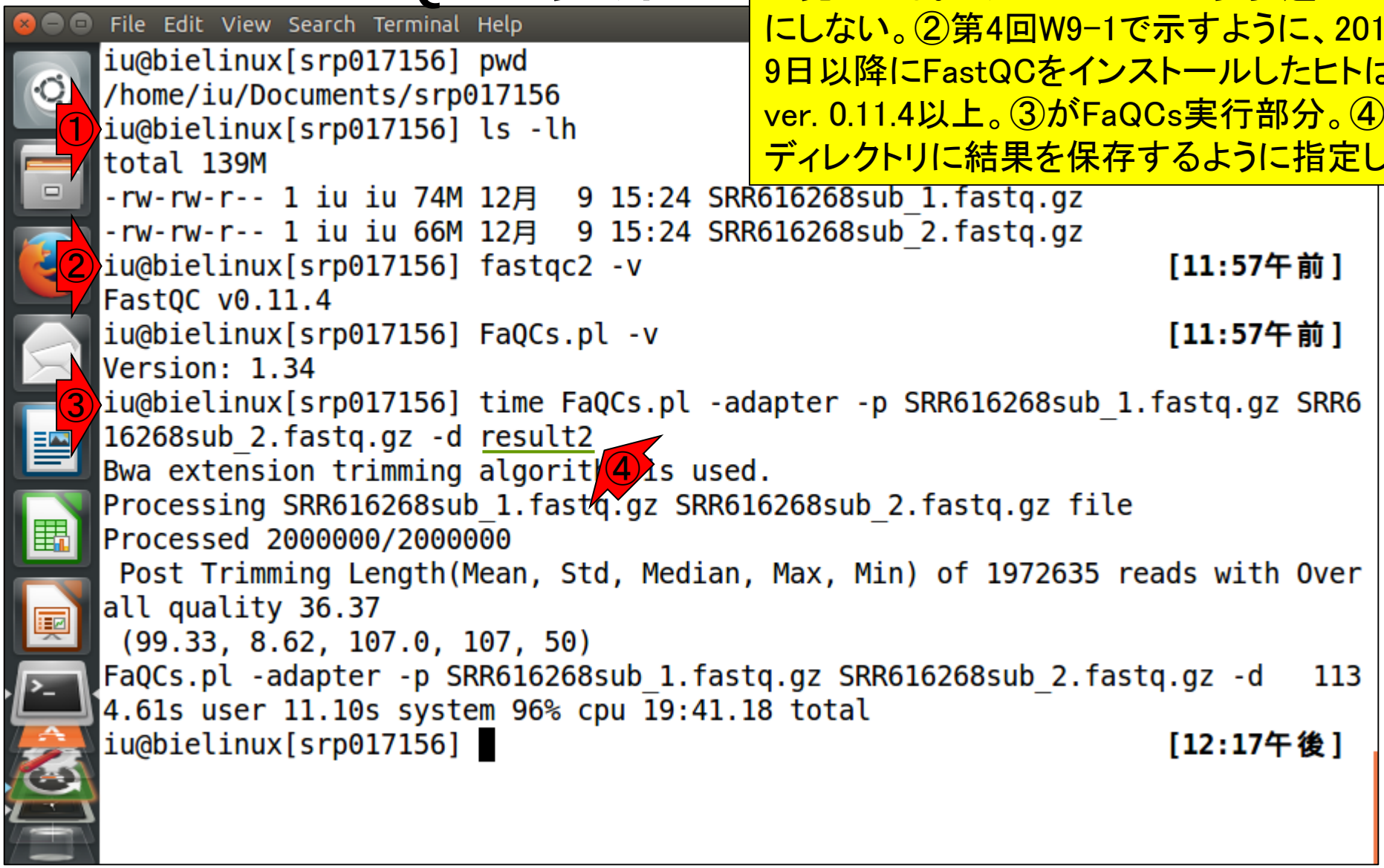
- Skewer: Jiang et al., BMC Bioinformatics, 2014
- FaQCs: Lo and Chain, BMC Bioinformatics, 2

- 図1。第4回の[W17-3]と基本的に同じ。[W1-1]

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

W1-1：FaQCs実行

第4回W17-3と基本的に同じ。①ls実行結果で入力として利用する2つのgzip圧縮ファイル（*.fastq.gz）が見られる。ファイルサイズが多少違っていても気にしない。②第4回W9-1で示すように、2015年10月9日以降にFastQCをインストールしたヒトは、ここがver. 0.11.4以上。③がFaQCs実行部分。④result2ディレクトリに結果を保存するように指定している



Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

おさらい（入力ファイル）

④まずは赤下線部分のFaQCs
の入力ファイルについておさらい

```
File  Edit  View  Search  Terminal  Help                          Ja  12:22
iu@bielinux[srp017156] pwd                                        [11:57午前]
/home/iu/Documents/srp017156
①iu@bielinux[srp017156] ls -lh                                    [11:57午前]
total 139M
-rw-rw-r-- 1 iu iu 74M 12月  9 15:24 SRR616268sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 66M 12月  9 15:24 SRR616268sub_2.fastq.gz
②iu@bielinux[srp017156] fastqc2 -v                                [11:57午前]
FastQC v0.11.4
iu@bielinux[srp017156] FaQCs.pl -v                                [11:57午前]
Version: 1.34
③iu@bielinux[srp017156] time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR6
16268sub_2.fastq.gz -d result2
Bwa extension trimming algorithm is used.
Processing SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz file
Processed 2000000/2000000
 Post Trimming Length(Mean, Std, Median, Max, Min) of 1972635 reads with Over
all quality 36.37
 (99.33, 8.62, 107.0, 107, 50)
FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d   113
4.61s user 11.10s system 96% cpu 19:41.18 total
iu@bielinux[srp017156]                                            [12:17午後]
```

④ ④ ④

おさらい（入力ファイル）

①乳酸菌NGS連載第5回のサイト。②のおさらい部分を軽く説明。場所は、さきほどのFaQCs実行コマンドの上部。スライドを見るだけ

(Rで)塩基配列解析

~NGS、RNA-seq、ゲノム、トランスクリプトーム、正規化、発現変動、統計、モ

(last m

What's

- このウ
  リーン
  法(W

- 書籍 | 日本乳酸菌学会誌 | について (last modified 2016/05/12) NEW
- 書籍 | 日本乳酸菌学会誌 | 第1回イントロダクション (last modified 2015/09/11)
- 書籍 | 日本乳酸菌学会誌 | 第2回GUI環境からコマンドライン環境へ (last modified 2
- 書籍 | 日本乳酸菌学会誌 | 第3回Linux環境構築からNGSデータ取得まで (last modi
- 書籍 | 日本乳酸菌学会誌 | 第4回クオリティコントロールとプログラムのインストール
- 書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC modified 2
- 書籍 | 日本乳酸菌学会誌 | 第6回ゲノムアセンブリ (last modified 2016 4/23)
- 書籍 | 日
- イントロ |

①

書籍 | 日本乳酸菌学会誌 | 第5回アセンブル、マッピング、そしてQC

日本乳酸菌学会誌の第5回分です。Linuxコマンドのリンク先は主に日経BP社様です。

- 第5回分PDF
- ウェブ資料PDF(201

Linuxコマンド

- bzip2 (bzip2圧縮、
- cd (ディレクトリを変
- echo (文字列を表
- export (変数を追加
- file (ファイルタイプ

②

乳酸菌(Lactobacillus casei 12A) paired-end RNA-seqデータのおさらい

Illumina HiSeq 2000で得られたデータです。

- 公共DB
  - SRR616268 (DDBJ SRA; DRA)
  - SRR616268 (EMBL-EBI ENA)
  - SRR616268 (NCBI SRA)

- DRAから取得したbzip2圧縮ファイル。2つとも行数は539,023,984行、リード数は134,755,996リード(約1.35億)。
  - forward側: SRR616268_1.fastq.bz2、7,662,128,101 bytes (約7.2GB)、全リード107 bp。
  - reverse側: SRR616268_2.fastq.bz2、7,017,031,734 bytes (約6.6GB)、全リード93 bp。

```
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
ls -l *.fastq.bz2
bzip2 -dc SRR616268_1.fastq.bz2 | wc
bzip2 -dc SRR616268_2.fastq.bz2 | wc
```

おさらい（入力ファイル）

①乳酸菌のpaired-end RNA-seqデータ。②Illumina HiSeq 2000で得られ、③オリジナルはそれぞれ約1.35億リード。④リードの長さは、forward側が107 bp、reverse側が93 bp。bzip2圧縮ファイル状態でも、計約14GBに達する!見るだけ

乳酸菌(Lactobacillus casei 12A) paired-end RNA-seqデータのおさらい ①

Illumina HiSeq 2000で得られたデータです。 ②

- 公共DB
  - SRR616268 (DDBJ SRA; DRA)
  - SRR616268 (EMBL-EBI ENA)
  - SRR616268 (NCBI SRA)

- DRAから取得したbzip2圧縮ファイル。2つとも行数は539,023,984行、リード数は134,755,996リード(約1.35億)。 ③
  - forward側: SRR616268_1.fastq.bz2、7,662,128,101 bytes (約7.2GB)、全リード107 bp。
  - reverse側: SRR616268_2.fastq.bz2、7,017,031,734 bytes (約6.6GB)、全リード93 bp。 ④

```
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
ls -l *.fastq.bz2
bzip2 -dc SRR616268_1.fastq.bz2 | wc
bzip2 -dc SRR616268_2.fastq.bz2 | wc
```

おさらい（入力ファイル）

①DDBJ SRA (DRA)から、②(#でコメントアウトしているが)FASTQファイルをダウンロードすると、bzip2圧縮ファイル(.bz2)として得られる。③は、元の.bz2ファイルを残しつつ、ファイルの行数をwcでカウントしている。④がその結果。⑤4で割って、リード数情報を得ている

乳酸菌(Lactobacillus casei 12A) paired-end RNA-seqデータのおさらい

Illumina HiSeq 2000で得られたデータです。

- 公共DB
  - ① SRR616268 (DDBJ SRA; DRA)
  - SRR616268 (EMBL-EBI ENA)
  - SRR616268 (NCBI SRA)

- DRAから取得したbzip2圧縮ファイル。2つとも行数は539,023,984行、リード数は134,755,996リード(約1.35億)。
  - forward側: SRR616268_1.fastq.bz2、7,662,128,101 bytes (約7.2GB)、全リード107 bp。
  - reverse側: SRR616268_2.fastq.bz2、7,017,031,734 bytes (約6.6GB)、全リード93 bp。

④ ⑤

```
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
#wget -c ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/SRA061/SRA061483/SRX2042
ls -l *.fastq.bz2
bzip2 -dc SRR616268_1.fastq.bz2 | wc
bzip2 -dc SRR616268_2.fastq.bz2 | wc
```

② ③

おさらい（入力ファイル）

①オリジナルはデカすぎるので、最初の
②4000000行分のみからなるサブセットを
③gzipした、④の.gzファイルが…

乳酸菌(Lactobacillus casei 12A) paired-end RNA-seqデータのおさらい

Illumina HiSeq 2000で得られたデータです。

- 公共DB
  - SRR616268 (DDBJ SRA; DRA)
  - SRR616268 (EMBL-EBI ENA)
  - SRR616268 (NCBI SRA)

- DRAから取得したbzip2圧縮ファイル。2つとも行数は539,023,984行、リード数は134,755,996リード(約1.35億)。
  - forward側: SRR616268_1.fastq.bz2、7,662,128,101 bytes (約7.2GB)、全リード107 bp。
  - reverse側: SRR616268_2.fastq.bz2、7,017,031,734 bytes (約6.6GB)、全リード93 bp。

```
#wget -c
#wget -c
ls -l *.
bzip2 -d
bzip2 -d
```

- サブセットの取得。2つとも行数は4,000,000行、リード数は1,000,000リード(100万)。
  連載第3回の[W25-2]あたり。
  - forward側: SRR616268sub_1.fastq.gz、74,906,576 bytes (約75MB)、全リード107 bp。
  - reverse側: SRR616268sub_2.fastq.gz、67,158,462 byt… (約67M…)、全リード93 bp。

②　③　④

①

```
bzip2 -dc SRR616268_1.fastq.bz2 | head -n 4000000 | gzip > SRR616268sub_1.fastq.gz
bzip2 -dc SRR616268_2.fastq.bz2 | head -n 4000000 | gzip > SRR616268sub_2.fastq.gz
```

- サブセットのgzip圧縮ファイルは以下のコピペでも取得可能。

```
wget -c http://www.iu.a.u-tokyo.ac.jp/~kadota/R_seq/SRR616268sub_1.fastq.gz
wget -c http://www.iu.a.u-tokyo.ac.jp/~kadota/R_seq/SRR616268sub_2.fastq.gz
ls -l *.fastq.gz
gzip -dc SRR616268sub_1.fastq.gz | wc
gzip -dc SRR616268sub_2.fastq.gz | wc
```

おさらい（入力ファイル）

③FaQCsの入力ファイルである、④SRR616268*.fastq.gzです

```
File  Edit  View  Search  Terminal  Help                    Ja  12:22
iu@bielinux[srp017156] pwd                                  [11:57午前]
/home/iu/Documents/srp017156
iu@bielinux[srp017156] ls -lh                               [11:57午前]
total 139M
-rw-rw-r-- 1 iu iu 74M 12月  9 15:24 SRR616268sub_1.fastq.gz
-rw-rw-r-- 1 iu iu 66M 12月  9 15:24 SRR616268sub_2.fastq.gz
iu@bielinux[srp017156] fastqc2 -v                           [11:57午前]
FastQC v0.11.4
iu@bielinux[srp017156] FaQCs.pl -v                          [11:57午前]
Version: 1.34
iu@bielinux[srp017156] time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR6
16268sub_2.fastq.gz -d result2
Bwa extension trimming algorithm is used.
Processing SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz file
Processed 2000000/2000000
 Post Trimming Length(Mean, Std, Median, Max, Min) of 1972635 reads with Over
all quality 36.37
 (99.33, 8.62, 107.0, 107, 50)
FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d   113
4.61s user 11.10s system 96% cpu 19:41.18 total
iu@bielinux[srp017156]                                      [12:17午後]
```

おさらい(FaQCs)

①FaQCs原著論文。是非使ってもらいたいというわけではなく、分散型バージョン管理システムGitHub上で公開されているプログラムの一例(連載第4回W15)として、またプログラム内部で用いるPerlモジュール(第4回W15)のインストール法など、これまで紹介してこなかったテクニックの伝授用としての意味合いが大きい

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQ
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
fastqc2 -q SRR616268sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

連載第4回[W9-5]の手順通りに行ったヒトは、fastqc2が FastQC ver. 0.11.3の実行プログラムになっているはずです。実行結果ファイルSRR616268sub_1_fastqc.html(forward側)では、(古いバージョンFastQC ver. 0.10.1の結果ですが)連載第4回W8-6に示すように、Overrepresented sequencesという項目中に「TruSeq Adapter, Index2と3」がリード中に含まれていることがわかります。また、SRR616268sub_2_fastqc.html (reverse側)では、「Illumina Single End PCR Primer 1」がリード中に含まれていることがわかります。

- Skewer: Jiang et al., BMC Bioinformatics, 2014
- FaQCs: Lo and Chain, BMC Bioinformatics, 2014 ①

- 図1。第4回の[W17-3]と基本的に同じ。[W1-1]

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

①

 Lo and Chain, *BMC Bioinformatics*, **15**: 366, 2014

おさらい（FaQCs）

FaQCs自体はQuality Control用プログラム。（平成27年度受講生の要望として挙がっていた）paired-endデータの①クオリティフィルタリングや、（特にIlluminaの）アダプター除去が-adapterオプションのみで簡単にできるので、精度に期待はしていなかったが使用感は思ったよりもよかった（個人の感想です）という代物です

BMC Bioinformatics. 2014 Nov 19;15:366. doi: 10.1186/s12859-014-0366-2.

Rapid evaluation and quality control of next genera

Lo CC[1], Chain PS[2,3].

Author information

Abstract

BACKGROUND: Next generation sequencing (NGS) technologies that parallelize the sequencing process and produce thousands to millions, or even hundreds of millions of sequences in a single sequencing run, have revolutionized genomic and genetic research. Because of the vagaries of any platform's sequencing chemistry, the experimental processing, machine failure, and so on, the quality of sequencing reads is never perfect, and often declines as the read is extended. These errors invariably affect downstream analysis/application and should therefore be identified early on to mitigate any unforeseen effects.

RESULTS: Here we present a novel FastQ Quality Control Software (FaQCs) that can rapidly process large volumes of data, and which improves upon previous solutions to monitor the quality and remove poor quality data from sequencing runs. Both the speed of processing and the memory footprint of storing all required information have been optimized via algorithmic and parallel processing solutions. The trimmed output compared side-by-side with the original data is part of the automated PDF output. We show how this tool can help data analysis by providing a few examples, including an increased percentage of reads recruited to references, improved single nucleotide polymorphism identification as well as de novo sequence assembly metrics.

CONCLUSION: FaQCs combines several features of currently available applications into a single, user-friendly process, and includes additional unique capabilities such as filtering the PhiX control sequences, conversion of FASTQ formats, and multi-threading. The original data and trimmed summaries are reported within a variety of graphics and reports, providing a simple way to do data quality control and assurance.

PMID: 25408143 [PubMed - indexed for MEDLINE] PMCID: PMC4246454 Free PMC Article

①FaQCsのGitHubサイト

おさらい（FaQCs）

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.3)を実行。
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
fastqc2 -q SRR616268sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

連載第4回[W9-5]の手順通りに行ったヒトは、fastqc2が FastQC ver. 0.11.3の実行プログラムになっているはずです。実行結果ファイルSRR616268sub_1_fastqc.html(forward側)では、(古いバージョンFastQC ver. 0.10.1の結果ですが)連載第4回W8-6に示すように、Overrepresented sequencesという項目中に「TruSeq Adapter, Index2と3」がリード中に含まれていることがわかります。また、SRR616268sub_2_fastqc.html (reverse側)では、「Illumina Single End PCR Primer 1」がリード中に含まれていることがわかります。

- Skewer: Jiang et al., BMC Bioinformatics, 2014
- FaQCs: Lo and Chain, BMC Bioinformatics, 2014

①

- 連載第4回の[W17-3]と基本的に同じ。[W1-1]

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

おさらい（FaQCs）

第4回W17-3。GitHubのページ中央あたりに、利用可能なオプションの説明あり。例えば①を眺めることで、-adapterオプションをつけられることを学ぶ。漫然と眺めるのではなく、FaQCsを通じてGitHubのページ構成や見方の経験を積むべし

GitHub | This repository Search | Explore Features Enterprise Blog | Sign up

LANL-Bioinformatics / FaQCs
forked from chienchi/FaQCs

Watch 5

45 commits | 1 branch | 1 release | 1 contributor

branch: master ▾ FaQCs / +

This branch is 23 commits ahead, 15 commits behind chienchi:master | Pull Request | Compare

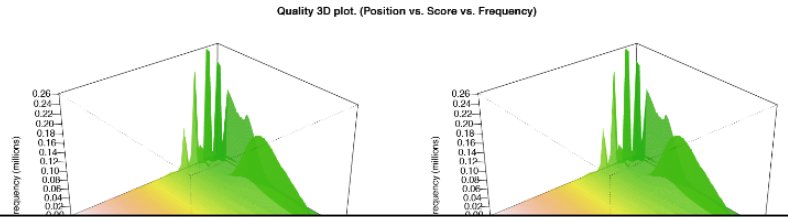
fix typo

chienchi authored on Mar 31 latest commit 6aed8354ec

example	adjust margin of pdf plots	3 months ago
galaxy_module	Update FaQCs.xml	6 months ago
lib	Add option "-5trim_off <bool> Turn off trimming from 5'end."	4 months ago
COPYRIGHT	add LICENSE	a year ago
FaQCs.pl	patch for jellyfish v2	3 months ago
LICENSE	add LICENSE	a year ago
README.md	fix typo	2 months ago

README.md

FaQCs: Quality Control of Next Generation Sequencing Data



BASIC

- Trimming by quality 5 and filtering reads with any ambiguous base or low complexity.

  $ perl FaQCs.pl -p reads1.fastq reads2.fastq -d out_directory

- Quailty check only on subsamples of input, no trimming and filtering.

  $ perl FaQCs.pl -p reads1.fastq reads2.fastq -d out_directory -qc_only

Full USAGE

```
Usage: perl FaQCs.pl [options] [-u unpaired.fastq] -p reads1.fastq reads2.fastq -d out_directory
Version 1.34
Input File: (can use more than once)
        -u              <Files> Unpaired reads

        -p              <Files> Paired reads in two files and separate by space
Trim:
        -mode           "HARD" or "BWA" or "BWA_plus" (default BWA_plus)
                        BWA trim is NOT A HARD cutoff! (see bwa's bwa_trim_read() function in bwa

        -q              <INT> Targets # as quality level (default 5) for trimming

        -5end           <INT> Cut # bp from 5 end before quality trimming/filtering

        -3end           <INT> Cut # bp from 3 end before quality trimming/filtering

        -adapter        <bool> Filter reads with illumina adapter/primers (default: no)
                        -rate   <FLOAT> Mismatch ratio of adapters' length (default: 0.2, allow 2

        -artifactFile  <File>    additional artifact (adapters/primers/contaminations) referenc
Filters:
        -min_L          <INT> Trimmed read should have to be at least this minimum length (defaul

        -avg_q          <NUM> Average quality cutoff (default:0, no filtering)
```

①

おさらい（FaQCs）

①今実行中のコマンドには-adapterオプションがついている。この経緯について説明。まず、FaQCsを最初に実行したとき（第4回W17-1）は、-adapterをつけていなかった

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.3)を実行。
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fastq.gz --outdir=/home/iu/Desktop/mac_share
fastqc2 -q SRR616268sub_2.fastq.gz --outdir=/home/iu/Desktop/mac_share
```

連載第4回[W9-5]の手順通りに行ったヒトは、fastqc2がFastQC ver. 0.11.3の実行プログラムになっているはずです。実行結果ファイルSRR616268sub_1_fastqc.html(forward側)では、(古いバージョンFastQC ver. 0.10.1の結果ですが)連載第4回W8-6に示すように、Overrepresented sequencesという項目中に「TruSeq Adapter, Index2と3」がリード中に含まれていることがわかります。また、SRR616268sub_2_fastqc.html (reverse側)では、「Illumina Single End PCR Primer 1」がリード中に含まれていることがわかります。

- Skewer: Jiang et al., BMC Bioinformatics, 2014
- FaQCs: Lo and Chain, BMC Bioinformatics, 2014

- 図1。第4回の[W17-3]と基本的に同じ。[W1-1]

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

①

おさらい（FaQCs）

–adapterをつけずにFaQCsを実行したFASTQファイルを入力として、FastQCを実行した結果（第4回W17–2）。①Overrepresented sequencesを眺めると、②（Illuminaの）TruSeq Adapter, Index 2が見つかった。このことからデフォルトではアダプター配列除去までは行えないことを学び、第4回W17-3で-adapterをつけてみるといいんじゃないかと学び、それをまずは最初に実行した（という次第）

FastQC Report

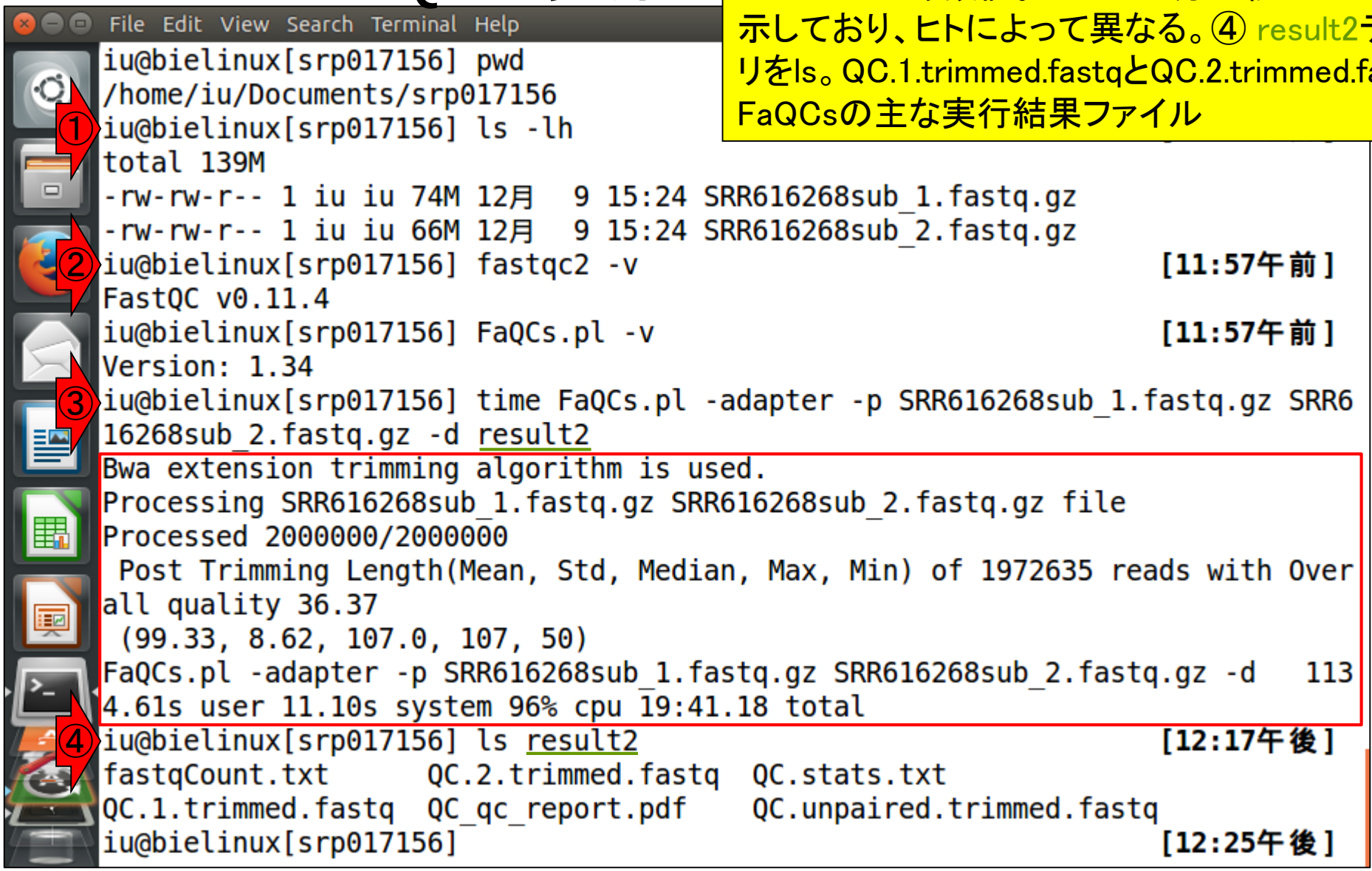
Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences ①
- Adapter Content
- Kmer Content

CTTTCGACAATGGACCTTATCGCTCACTGTC			
ATTTAGCCTTGGGAGATGGTCCTCCCGGATT			
CGCTACTCATGCCGGCATTCTCACTTCTAAG			
CCGGGGTGCTTTTCACCTTTCCCTCACGGTACTGGTTCACTATCGGTCAC	1349	0.13802869646272048	No Hit
GGGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAAG	1341	0.13721014229541006	No Hit
CTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTG	1336	0.13669854594084105	No Hit
GCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTACT	1319	0.13495911833530638	No Hit
GCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGATACTCTCCTC	1305	0.13352664854251314	No Hit
AGATCGGAAGAGCACACGTCTGAACTCCAGTCACCGATGTATCTCGTATG	1296	0.13260577510428892	TruSeq Adapter, Index 2 (100% over 49bp) ②
GTTTGGGCTCTTCCCACTTCGCTCGCCGCTACTATGGGAATCGAGTTTTC	1262	0.12912691989321962	No Hit
CGTCCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTG	1258	0.12871764280956438	No Hit
CACACGGTTTCAGGAACTGTTTCACTCCCCTTCCGGGGTGCTTTTCACCT	1257	0.1286153235386506	No Hit
CCCTAGTTCAAACAGTGCTCTACCTCCATGATCCATCCTCCGAGGCTAAC	1249	0.12779676937134016	No Hit
GTCATTTTGCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGATA	1249	0.12779676937134016	No Hit
ACCTTAACTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCG	1223	0.1251364683275813	No Hit
CACAGTTTCGGTATTATGCTTAGCCCCGGTATATTTTCGGCGCAGTGCCA	1220	0.12482951051483988	No Hit
GGTCATTTTGCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGAT	1208	0.12360167926387423	No Hit
CCACTTAGCATAAATTTAGGGACCTTAACTGGTGATCTGGGCTGTTCCCC	1207	0.12349935999296043	No Hit
CCGGTTCATTCTACAAAAGGCACGCCATTACCCGTTAACGGGCTTTGACT	1191	0.12186225165833958	No Hit
CTGCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTA	1178	0.12053210113646014	No Hit

W1-1：FaQCs実行

③最初にコピペ実行したFaQCsの実行時間は、約20分。画面上で見えているのは、赤枠内の一番下の19:41.18という数値。これは19分41秒かかったことを示しており、ヒトによって異なる。④ result2ディレクトリをls。QC.1.trimmed.fastqとQC.2.trimmed.fastqがFaQCsの主な実行結果ファイル



Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

W1-2：FastQC実行

①左上と右下のスクリーンショットの重複（のりしろ）部分。②FastQCコピペ実行用コード。全部で6行分あるが、必要最小限は赤枠真ん中のfastqc2から始まる2行分だけ。それゆえpwd, ls, dateなどのコマンド実行結果はイチイチ説明しないが、「今どこで作業をしていて、入力ファイルがあるかどうか」の確認は自分でやっておこう

FaQCs(ver. 1.34)によるQC

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fa
fastqc2 -q SRR616268sub_2.fa
```

連載第4回[W9-5]の手順通りに行った
です。実行結果ファイルSRR616268su
結果ですが)連載第4回W8-6に示すよ
と3」がリード中に含まれていることが
「Illumina Single End PCR Primer 1」が

- Skewer: Jiang et al., BMC Bioinformat
- FaQCs: Lo and Chain, BMC Bioinform

- 図1。第4回の[W17-3]と基本的に同じ。

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SR
ls result2
```

①

```
pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

FaQCs ver. 1.34を実行し、結果をresult2ディレクトリに保存(約25分)。result2ディレクトリ上には、計6ファイルが生成されている。そのうちのQC結果のサマリーレポートは次の2つ：QC.stats.txtとQC_qc_report.pdf

- トリム後(FaQCs実行後)のデータを入力としてFastQC (ver. 0.11.3)をデフォルトで実行。[W1-2]

②

```
pwd
ls result2/*.fastq
fastqc2 -q result2/QC.1.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
fastqc2 -q result2/QC.2.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
date
ls -lh /home/iu/Desktop/mac_share/QC.*
```

出力結果を共有フォルダ(/home/iu/Desktop/mac_share)に直接保存。実行結果ファイルQC.1.trimmed_fastqc.html(forward側)とQC.2.trimmed_fastqc.html(reverse側)ともに、Overrepresented sequences項目に見えていたアダプターやプライマー配列情報がなくなっているのがわかります。[W1-3]

- Rockhopper 2: Tjaden B, Genome Biol., 2015
- QuasR: Gaidatzis et al., Bioinformatics, 2015

トランスクリプトームアセンブリ

W1-2：FastQC実行

以後は徐々に冗長な説明を省略していくが、③のW1-2のようなウェブ資料番号情報を頼りにして、自分でコピペ用コードを探して有効利用してください。Bio-Linux上でコピペできないなどのバグに遭遇して再起動に時間がかかることなどもあるでしょうが、コピペを有効利用すれば追いつけます。④の部分をコピペしたのが次のスライド

③

FaQCs(ver. 1.34)によるQ

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fa
fastqc2 -q SRR616268sub_2.fa
```

連載第4回[W9-5]の手順通りに行った
です。実行結果ファイルSRR616268su
結果ですが)連載第4回W8-6に示すよ
と3」がリード中に含まれていることが
「Illumina Single End PCR Primer 1」が

- Skewer: Jiang et al., BMC Bioinformat
- FaQCs: Lo and Chain, BMC Bioinform

- 図1。第4回の[W17-3]と基本的に同じ。

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SR
ls result2
```

```
pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

FaQCs ver. 1.34を実行し、結果をresult2ディレクトリに保存(約25分)。result2ディレクトリ上には、計6ファイルが生成されている。そのうちのQC結果のサマリーレポートは次の2つ：QC.stats.txtとQC_qc_report.pdf

- トリム後(FaQCs実行後)のデータを入力としてFastQC (ver. 0.11.3)をデフォルトで実行。[W1-2]

④

③

```
pwd
ls result2/*.fastq
fastqc2 -q result2/QC.1.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
fastqc2 -q result2/QC.2.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
date
ls -lh /home/iu/Desktop/mac_share/QC.*
```

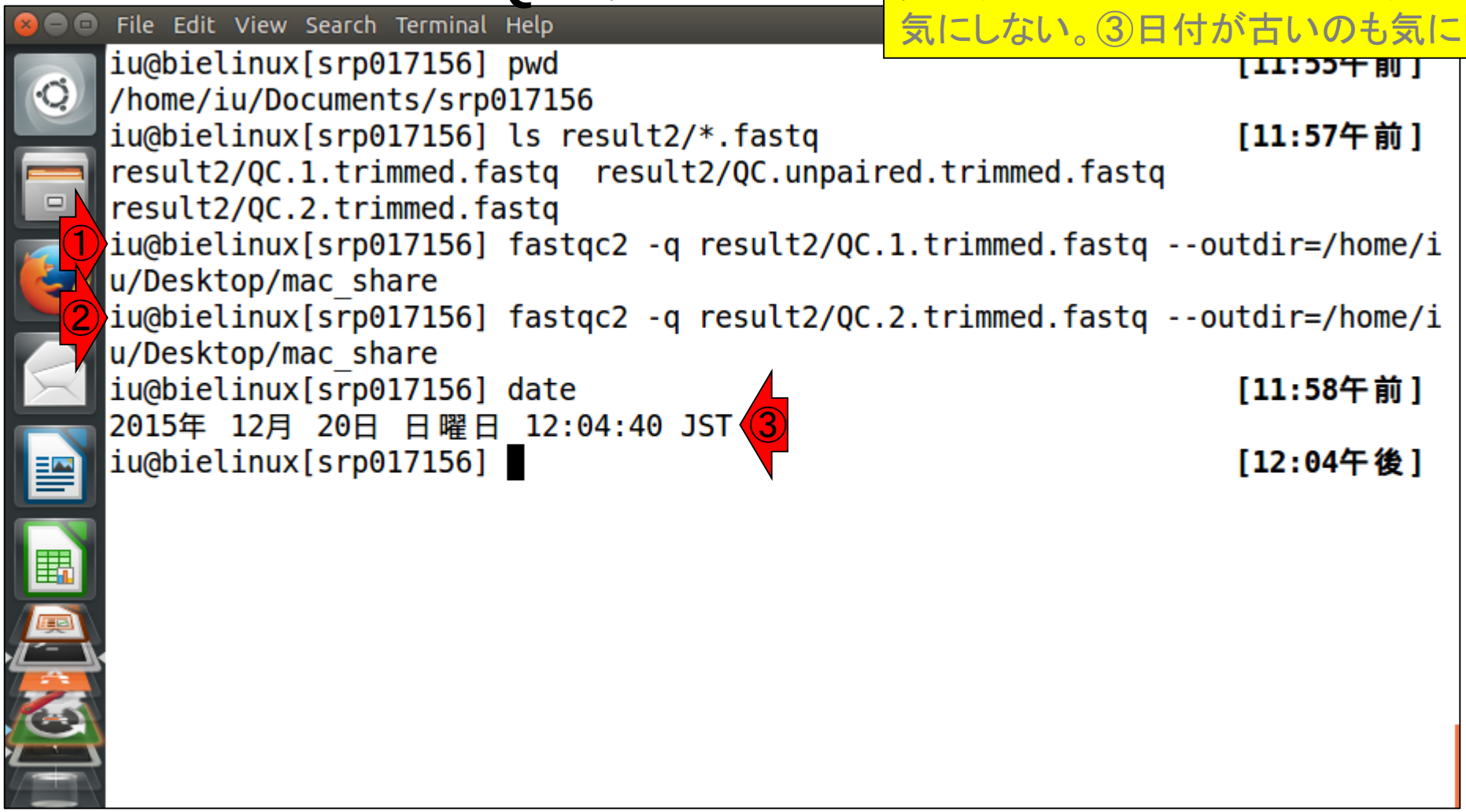
出力結果を共有フォルダ(/home/iu/Desktop/mac_share)に直接保存。実行結果ファイルQC.1.trimmed_fastqc.html(forward側)とQC.2.trimmed_fastqc.html(reverse側)ともに、Overrepresented sequences項目に見えていたアダプターやプライマー配列情報がなくなっているのがわかります。[W1-3]

- Rockhopper 2: Tjaden B, Genome Biol., 2015
- QuasR: Gaidatzis et al., Bioinformatics, 2015

トランスクリプトームアセンブリ

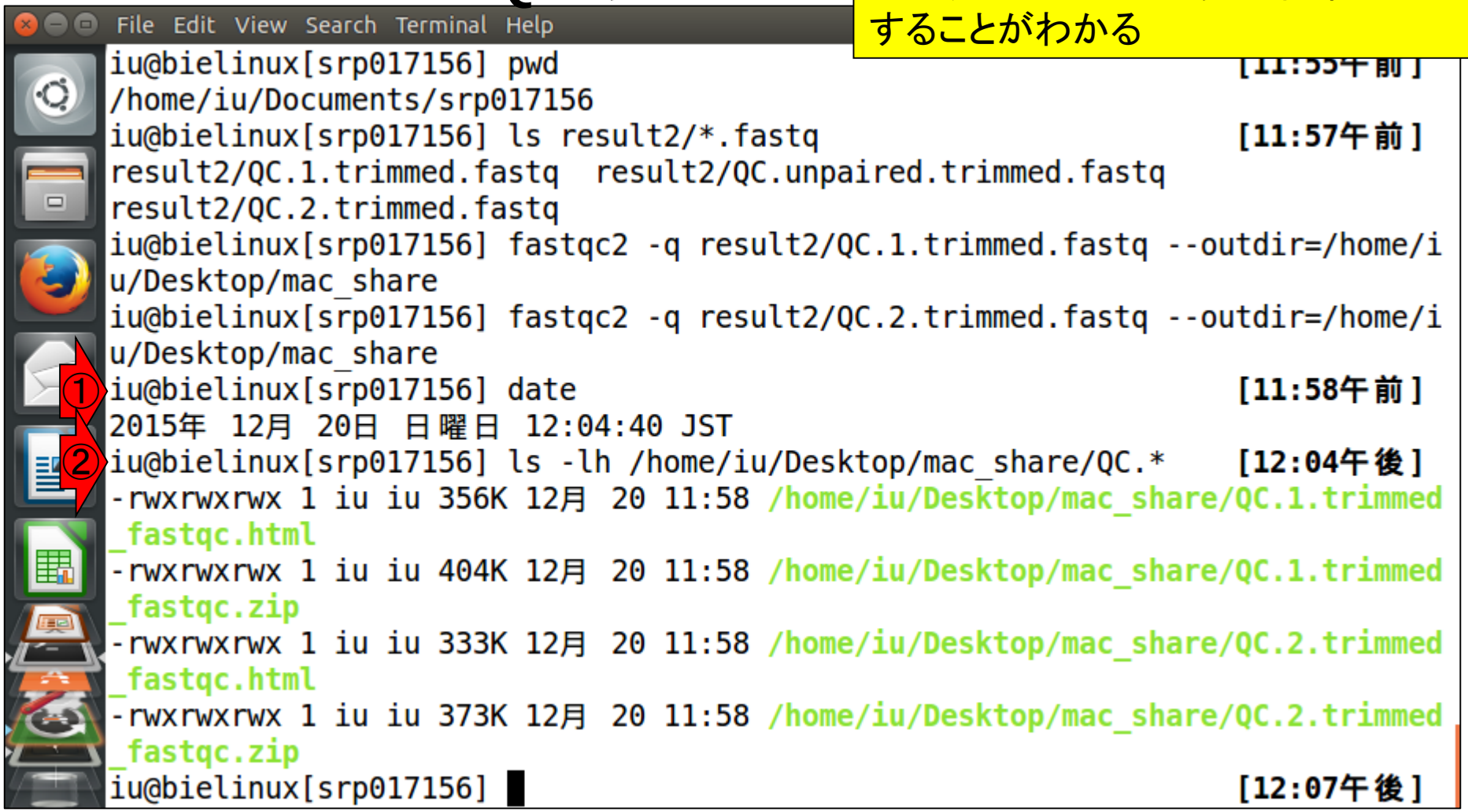
W1-2：FastQC実行

①forward側、②reverse側のFaQCs実行結果ファイルをFastQC (ver. 0.11.4)の入力として実行。ver. 0.11.3 or 0.11.4かの細かい違いは気にしない。③日付が古いのも気にしない



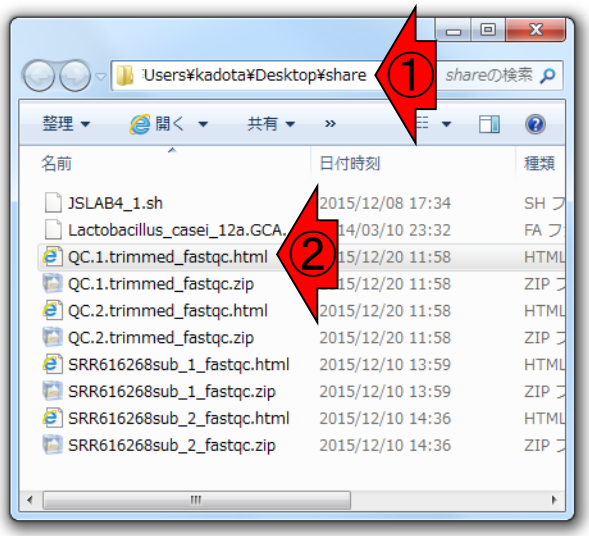
W1-2：FastQC実行

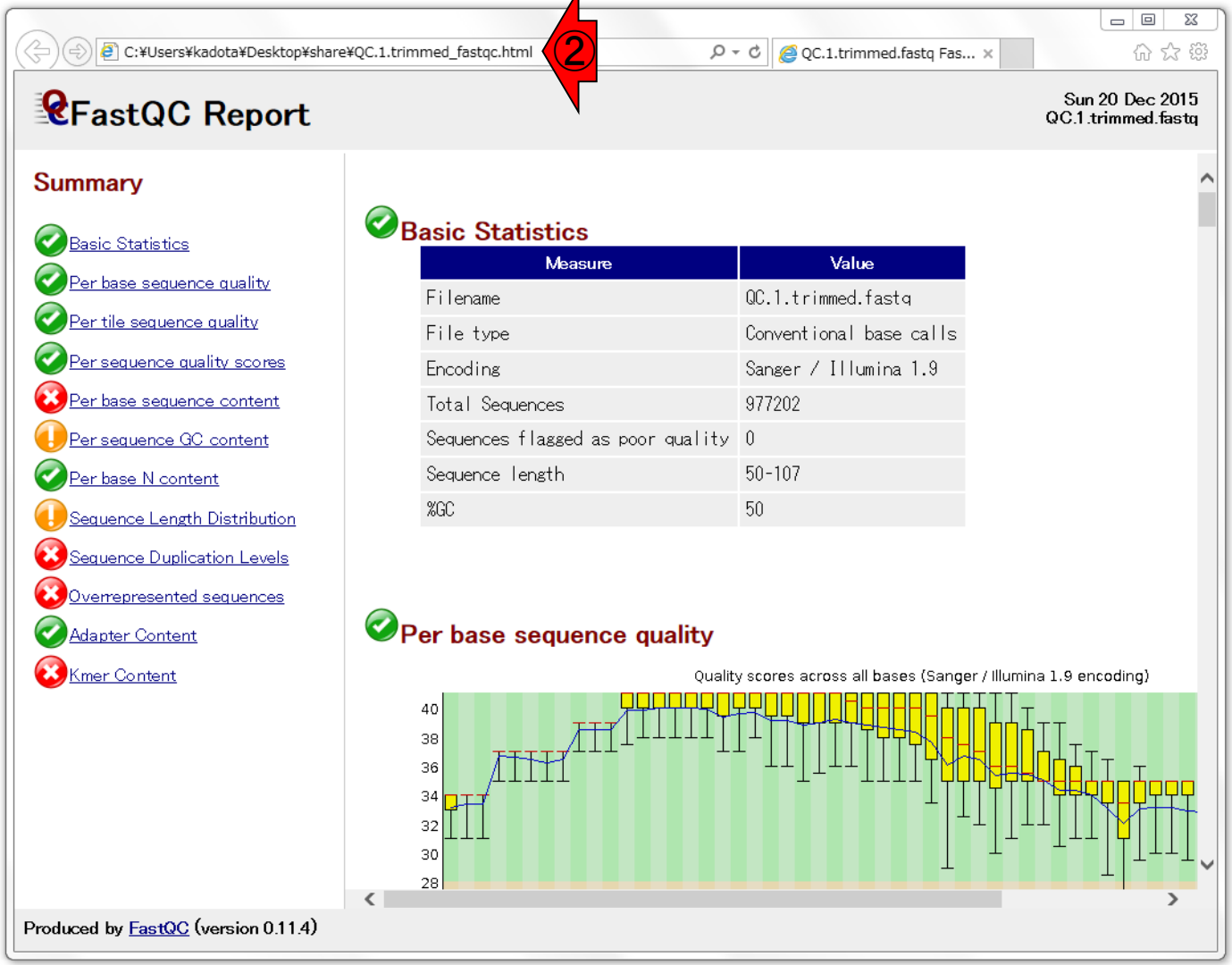
①dateで日付を表示。②保存先として指定した共有フォルダ(/home/iu/Desktop/mac_share)中に、確かにFastQC実行結果ファイルが存在することがわかる



W1-3：FastQCで確認

①ホストOS上の共有フォルダshare上で、②forward側のFastQC実行結果ファイル(QC.1.trimmed_fastqc.html)を眺める





W1-3：FastQCで確認

①W1-3関連は、赤枠で示すように、もしfastqc2実行できなくても、②（できたことにして）htmlの結果ファイルを眺めることはできます。こんな感じで結果ファイルを見られる場合もありますので、最終手段として適宜ご利用下さい

FaQCs(ver. 1.34)によるQ

①

- 連載第5回[W1-1]結果との比較用に、FaQCs実行前のデータでFastQC (ver. 0.11.
  連載第4回の[W7, W8, W9-7]あたり。

```
fastqc2 -v
fastqc2 -q SRR616268sub_1.fa
fastqc2 -q SRR616268sub_2.fa
```

連載第4回[W9-5]の手順通りに行った
です。実行結果ファイルSRR616268su
結果ですが)連載第4回W8-6に示すよ
と3」がリード中に含まれていることが
「Illumina Single End PCR Primer 1」が

- Skewer: Jiang et al., BMC Bioinformat
- FaQCs: Lo and Chain, BMC Bioinform

- 図1。第4回の[W17-3]と基本的に同じ。

```
cd ~/Documents/srp017156
rm -f hoge*
rm -f JS*
rm -rf result*
rm -f *.bz2

pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SR
ls result2
```

```
pwd
ls -lh
fastqc2 -v
FaQCs.pl -v
time FaQCs.pl -adapter -p SRR616268sub_1.fastq.gz SRR616268sub_2.fastq.gz -d result2
ls result2
```

FaQCs ver. 1.34を実行し、結果をresult2ディレクトリに保存(約25分)。result2ディレクトリ上には、計6ファイルが生成されている。そのうちのQC結果のサマリーレポートは次の2つ：QC.stats.txtとQC_qc_report.pdf

- トリム後(FaQCs実行後)のデータを入力としてFastQC (ver. 0.11.3)をデフォルトで実行。[W1-2]

```
pwd
ls result2/*.fastq
fastqc2 -q result2/QC.1.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
fastqc2 -q result2/QC.2.trimmed.fastq --outdir=/home/iu/Desktop/mac_share
date
ls -lh /home/iu/Desktop/mac_share/QC.*
```

②

出力結果を共有フォルダ(/home/iu/Desktop/mac_share)に直接保存。実行結果ファイル
QC.1.trimmed_fastqc.html(forward側)とQC.2.trimmed_fastqc.html(reverse側)ともに、Overrepresented sequences項目に見えていたアダプターやプライマー配列情報がなくなっているのがわかります。[W1-3]

①

- Rockhopper 2: Tjaden B, Genome Biol., 2015
- QuasR: Gaidatzis et al., Bioinformatics, 2015

トランスクリプトームアセンブリ

W1-3：FastQCで確認

①Overrepresented sequencesを眺めて Possible Sourceのところが全てNo Hitになっているかどうかをチェック

FastQC Report

Sun 20 Dec 2015
QC.1.trimmed.fastq

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences ①
- Adapter Content
- Kmer Content

Overrepresented sequences

②

Sequence	Count	Percentage	Possible Source
CCCCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCA	14289	1.4622360576421252	No Hit
GGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAAGT	10984	1.124025534127028	No Hit
GTGCTTTTCACCTTTCCCTCACGGTACTGGTTCACTATCGGTCACTAGGG	8746	0.8950043082187716	No Hit
CCCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCAC	8446	0.8643044119844209	No Hit
GCCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCGACCTTC	8081	0.8269528715659608	No Hit
GTCACTAGGGAGTATTTAGCCTTGGGAGATGGTCCTCCCGGATTCCGACG	7943	0.8128309192981594	No Hit
GTCCAGTCCTACAACCCCGAGAAGCAAGCTTCTCGGTTTGGGCTCTTCCC	6618	0.6772397109297771	No Hit
GTCGGTTTGCGGTACGGGTAGTTTATTTCTCACTAGAAGCTTTTCTTGGC	6373	0.6521681290050573	No Hit
GGTCACTTGGTTTCGGGTCTACATCTGCTTACTCATTCGCCCTGTTCAGA	5461	0.558840444452631	No Hit
GCCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCAACCTTC	4804	0.491607671699403	No Hit
CCGGTATATTTTCGGCGCAGTGCCACTCGACTAGTGAGCTATTACGCACT	4378	0.44801381904662496	No Hit
CCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTGCTT	4342	0.44432983149850286	No Hit
CCCGCGTCTGCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTG	4335	0.443613500586368	No Hit
CCACAGTTTCGGTATTATGCTTAGCCCCGGTATATTTTCGGCGCAGTGCC	4292	0.4392131821261111	No Hit
CTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTGTCTGACTC	4087	0.41823491969930476	No Hit
CCGGCATTCTCACTTCTAAGCGCTCCAGCCGTCCTCACGATCGACCTTCA	3839	0.39285633881224147	No Hit

W1-3：FastQCで確認

①連載第4回W17-2で見られたアダプター配列がなくなっていることがわかる。これを含め、全てNo Hitになっていたことから、-adapterオプションがうまく機能していることがわかる。同じ手順で、reverse側のFastQC実行結果ファイル(QC.2.trimmed_fastqc.html)も眺めておこう

FastQC Report

Summary

- Basic Statistics
- Per base sequence quality
- Per tile sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Sequence Duplication Levels
- Overrepresented sequences
- Adapter Content
- Kmer Content

CGCTACTCATGCCGGCATTCTCACTTCTAAC				
CCGGGGTGCTTTTCACCTTTCCCTCACGGTACTGGTTCACTATCGGTCAC	1349	0.13804720006713045	No Hit	
GGGCCTATTCACTGCGGCTGACCTTGCGGTCAGCACCCCTTCTTCCGAAG	1341	0.1372853616754777	No Hit	
CTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCGCTCACTG	1336	0.13671687123030857	No Hit	
GCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTACT	1319	0.13497721044369537	No Hit	
① GCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGATACTCTCCTC	1305	0.13354454861942566	No Hit	
GTTTGGGCTCTTCCCACTTCGCTCGCCGCTACTATGGGAATCGAGTTTTC	1262	0.12914423015916873	No Hit	
CGTCCCTCCATCGCTTAAACAAAATAAACTAGTGCAGGAATCTCAACCTG	1258	0.1287348982093774	No Hit	
CACACGGTTTCAGGAACTGTTTCACTCCCCTTCCGGGGTGCTTTTCACCT	1257	0.12863256522192956	No Hit	
CCCTAGTTCAAACAGTGCTCTACCTCCATGATCCATCCTCCGAGGCTAAC	1249	0.12781390132234688	No Hit	
GTCATTTTGCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGATA	1249	0.12781390132234688	No Hit	
ACCTTAACTGGTGATCTGGGCTGTTCCCCTTTCGACAATGGACCTTATCG	1223	0.12515324364870312	No Hit	
CACAGTTTCGGTATTATGCTTAGCCCCGGTATATTTTCGGCGCAGTGCCA	1220	0.12484624468635963	No Hit	
GGTCATTTTGCCGAGTTCCTTAACGAGAGTTCGCTCGCTCACCTGAGGAT	1208	0.1236182488369856	No Hit	
CCACTTAGCATAAATTTAGGGACCTTAACTGGTGATCTGGGCTGTTCCCC	1207	0.12351591584953776	No Hit	
CCGGTTCATTCTACAAAAGGCACGCCATTACCCGTTAACGGGCTTTGACT	1191	0.1218785880503724	No Hit	
CTGCCGCCGGCCAGCTATGTATTCACTGACAAGCAATACACTGATGTGTA	1178	0.12054825921355053	No Hit	
CCCCTACTCACCATCCTCCACCCACCCTTCCACCTTTCCCTTACACCCCT	1174	0.12012892726375917	No Hit	

Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

第5回原稿PDF

ここまでで、第5回原稿PDFの①p195の②左上あたりまでを述べた。FaQCs実行によって、FastQCのOverrepresented sequences項目で見えていたIllumina adapterが消えたので第一段階クリアということで安心している状態

として、FastQC によるクオリティチェックを行えばよい［W1-2］。著者らは、FastQC 実行結果ファイルの項目（Overrepresented sequences）を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している［W1-3］。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ（assembly）であり、再構築を行うプログラムをアセンブラ（assembler）という。デノボ（*de novo*）という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」という意味
入力として（つまり他の
レする際には、*de novo*
トランスクリプトームアセンブリとは、アセンブル対象がゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブルする場合は、*de novo* transcriptome assembly などと呼ばれる。

Multiple-k [20] や Trans-ABySS [21] などの初期のトランスクリプトーム用アセンブラは、ゲノム用を内部的に用いていた。詳細は省くが、上述の k-mer の k の値（正の整数）を大きくすればするほど、得られるコンティグは長くなり、高発現のものに偏る傾向にある [22]。k の値は、アセンブル時の「のりしろ」に相当するものである。パリンドロームを避けるべく、通常は奇数が採用される [23]。k の値を小さくするほど、低発現転写物を拾いあげることが原理的には可能であるが、得られるコンティグは短くなり（断片化）、似た配列からなるコンティグが多く得られる傾向（重複）にある。このためこれらのプログラムは、複数の k の値を用いて独立にゲノム用アセンブラを適用し、できるだけ多くの転写物配列をコンティグとして得ることに主眼を置いていた。それゆえ、コンティグ集合からいかに重複をとり除くかが課題であった。

おそらく現在もっとも頻用されているトランスクリプトーム用アセンブラは、Trinity [24] である。Trinity は、トランスクリプトーム専用としてデザインされた最初のプ

②

①

第5回原稿PDF

赤枠の最初のほうでオチを先に述べているが、当時は、この乳酸菌RNA-seqデータの次の解析手段として、バクテリア用*de novo*トランスクリプトームアセンブラである①Rockhopper2の論文を発見した。バクテリア用だから、有名なTrinityなどよりもさぞかしよりよい結果が得られるだろうと思っていた

として、FastQC によるクオリティチェックを行えばよい［W1-2］。著者らは、FastQC 実行結果ファイルの項目（Overrepresented sequences）を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している［W1-3］。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ（assembly）であり、再構築を行うプログラムをアセンブラ（assembler）という。デノボ（*de novo*）という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」とい
入力として（つま
ルする際には、*de*
トランスクリプト
ゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブルする場合は、*de novo* transcriptome assembly などと呼ばれる

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

①

W2-1：Rockhopper

①このあたり。トランスクリプトーム用*de novo*アセンブラの②Trinityと③Rockhopper2。ここでは、トランスクリプトームアセンブリというよりも、Javaプログラムの1つとしてのRockhopper2をどうやって実行させるかを紹介するという位置づけ

トランスクリプトームアセンブリ ①

- Multiple-k: Surget-Groba and Montoya-Burgos, Genome Res., 20
- Trans-ABySS: Robertson et al., Nat Methods, 2010
- Gibbons et al., Mol Biol Evol., 2009
- Miller et al., Genomics, 2010

- Trinity: Grabherr et al., Nat Biotechnol, 2011 ②
- Bridger: Chang et al., Genome Biol., 2015

- CD-HIT: Fu et al., Bioinformatics, 2012
- IFRAT: Mbandi et al., BMC Bioinformatics, 2015

- Cufflinks: Trapnell et al., Nat Biotechnol., 2010

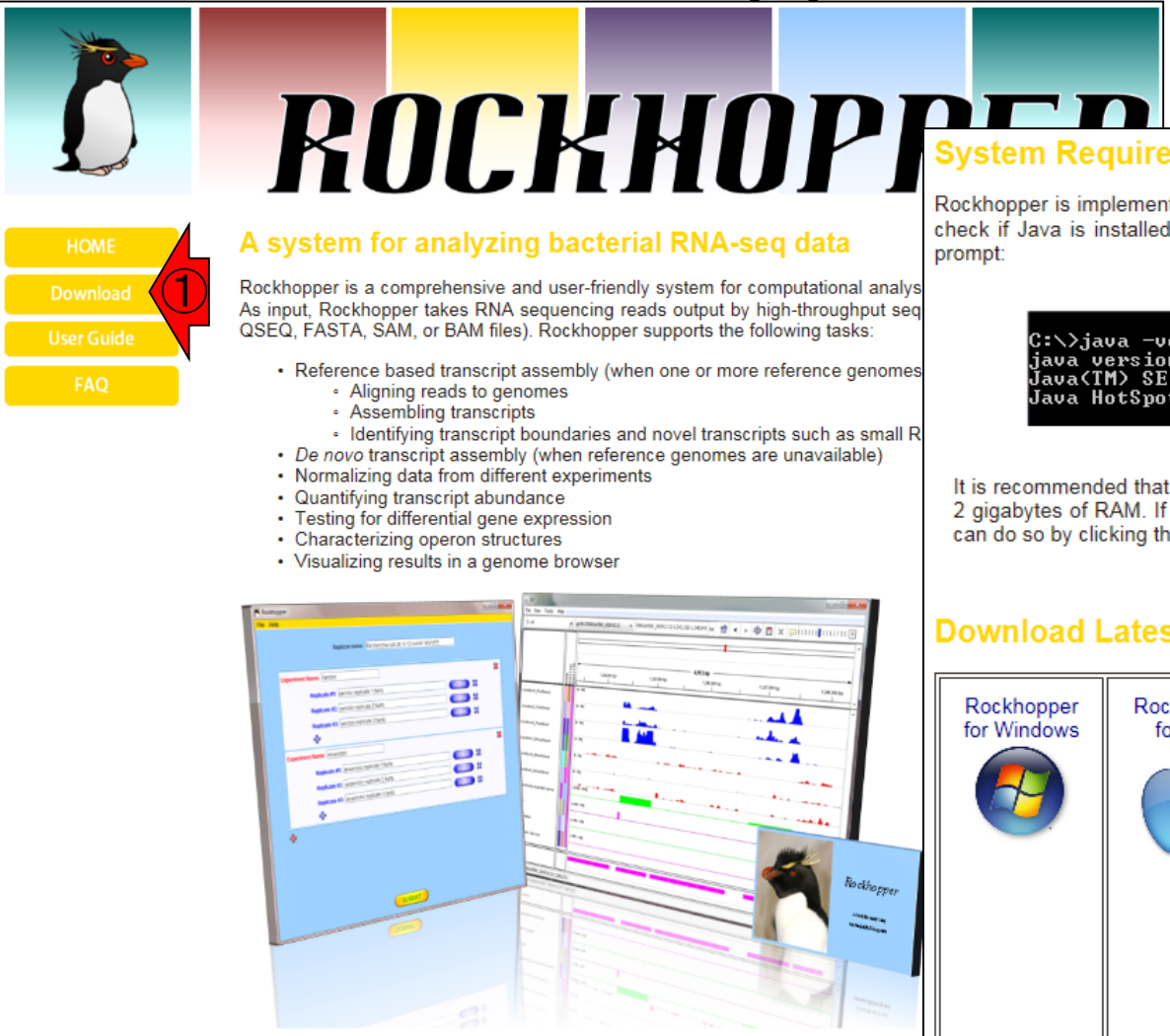
- Rockhopper(バクテリア用): McClure et al., Nucleic Acids Res., 2013
- TruHMM(バクテリア用): Li et al., BMC Genomics, 2013
- Rockhopper 2(バクテリア用): Tjaden B, Genome Biol., 2015 ③

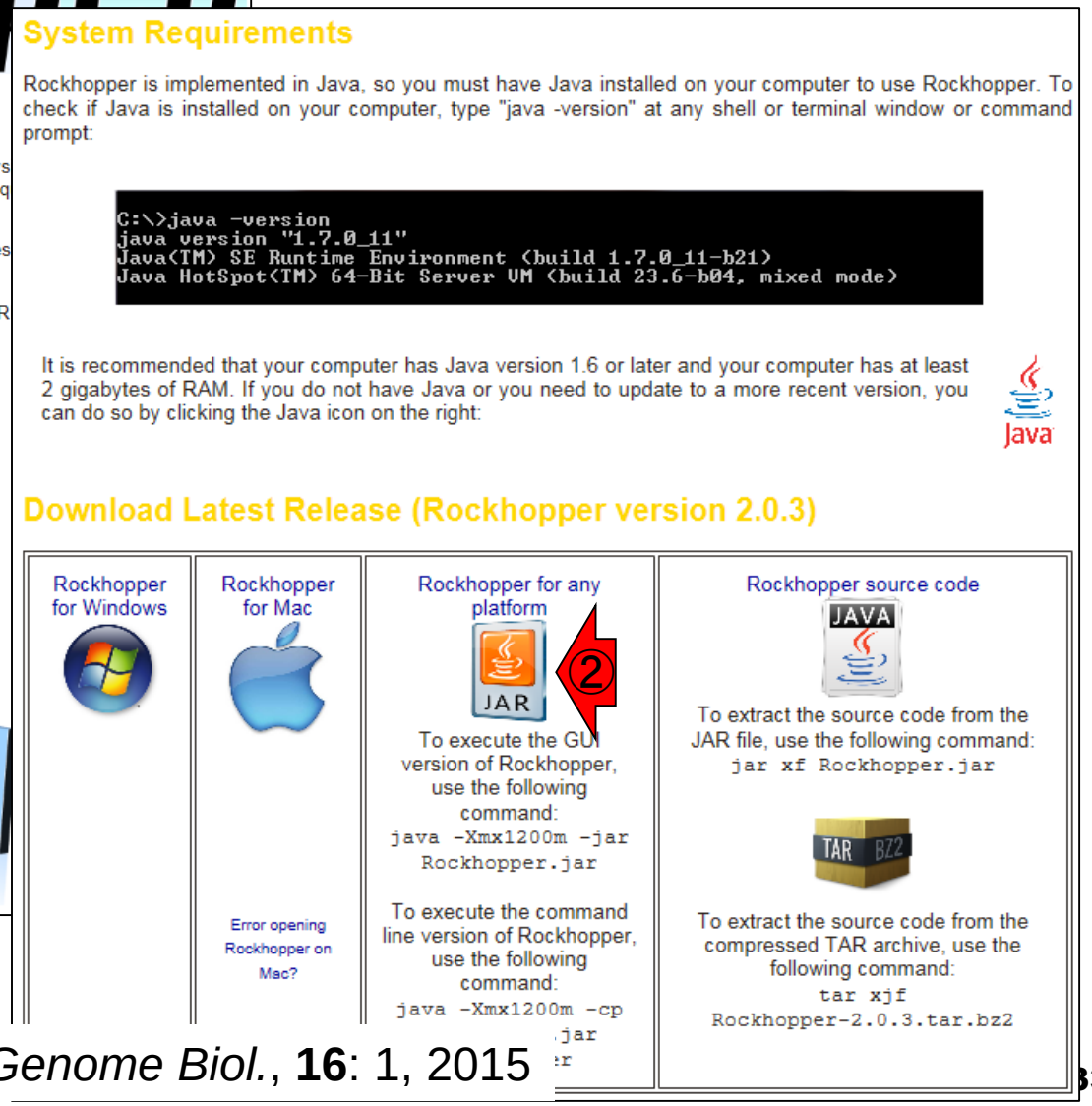
De novo transcriptome assembly (Rockhopper2 ver. 2.0.3)

- Rockhopper2(バクテリア用): Tjaden B, Genome Biol., 2015 ③
- Java確認など[W2-2]

W2-1：Rockhopper

①Download。Bio-Linux8の場合は、②「Rockhopper for any platform」でよい。実際にはダウンロード済みなのでエアーハンズオン（やったつもり）

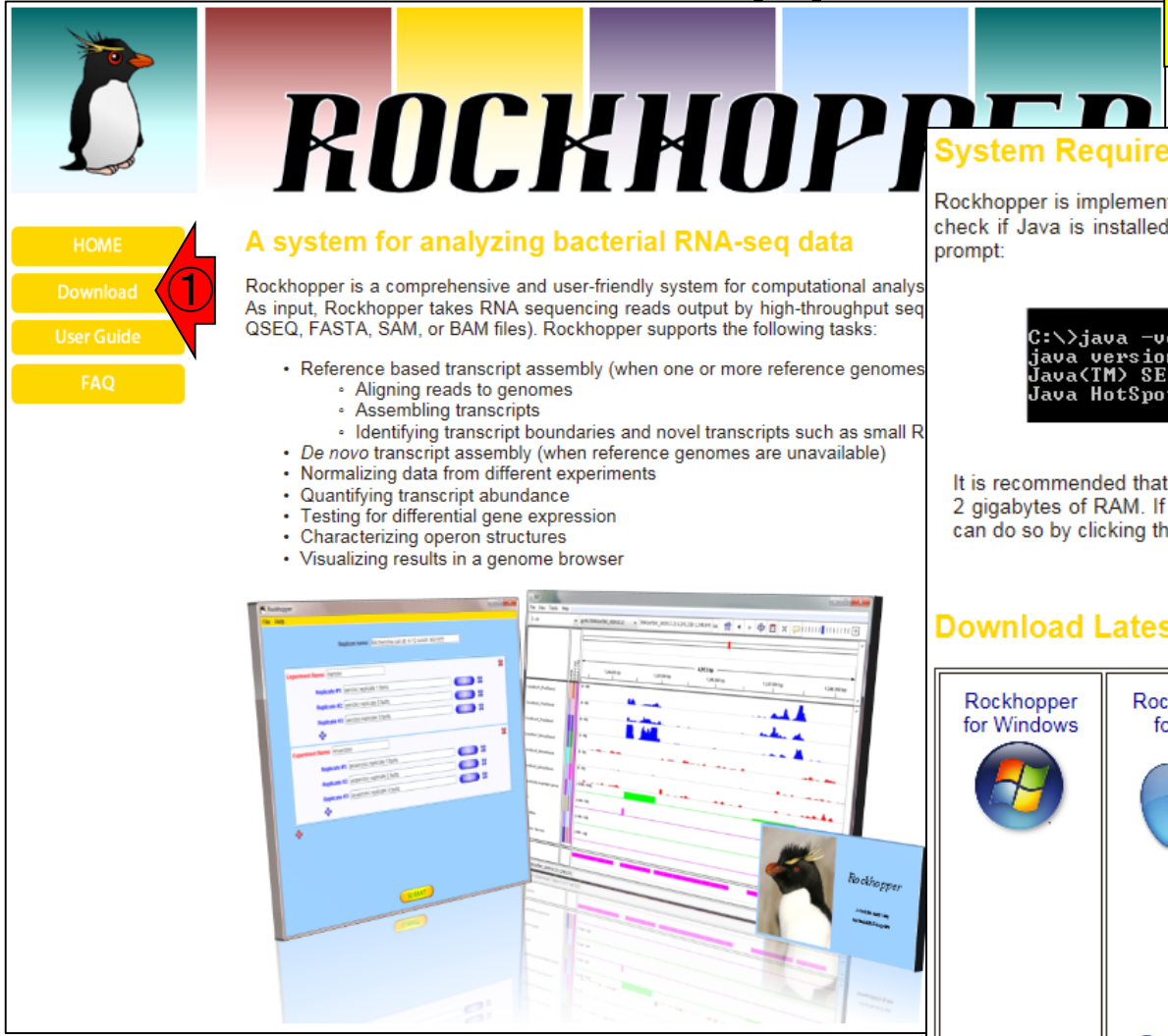




Tjaden, B., *Genome Biol.*, **16**: 1, 2015

W2-1 : Rockhopper

基本的には右クリックで②「対象をファイルに保存」でよいが、wgetコマンドを利用したい場合は③「ショートカットのコピー（Windowsの場合）」でURL情報を取得する



W2-1：Rockhoppe

①System Requirements。これはFaQCsのときのPrerequisite（前もって必要な事柄）と似たようなもの。Rockhopper2はJavaプログラムなので、Javaがver. 1.6以上かどうかを②「java -version」で確認(次スライド)

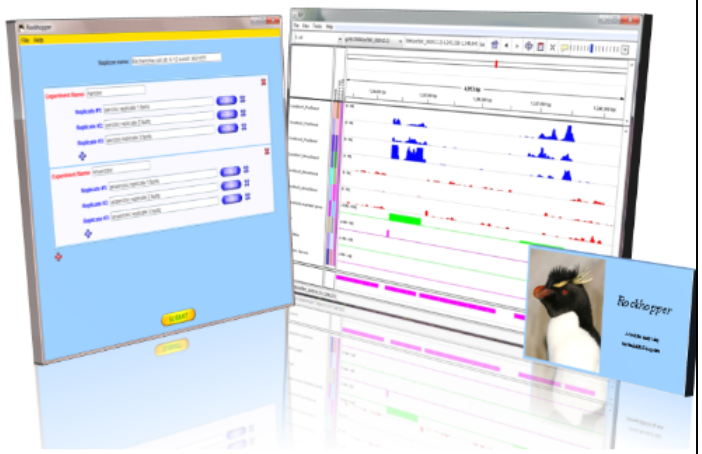
ROCKHOPPER

HOME
Download
User Guide
FAQ

A system for analyzing bacterial RNA-seq data

Rockhopper is a comprehensive and user-friendly system for computational analys
As input, Rockhopper takes RNA sequencing reads output by high-throughput seq
QSEQ, FASTA, SAM, or BAM files). Rockhopper supports the following tasks:

- Reference based transcript assembly (when one or more reference genomes
  - Aligning reads to genomes
  - Assembling transcripts
  - Identifying transcript boundaries and novel transcripts such as small R
- *De novo* transcript assembly (when reference genomes are unavailable)
- Normalizing data from different experiments
- Quantifying transcript abundance
- Testing for differential gene expression
- Characterizing operon structures
- Visualizing results in a genome browser



System Requirements

①

Rockhopper is implemented in Java, so you must have Java installed on your computer to use Rockhopper. To check if Java is installed on your computer, type "java -version" at any shell or terminal window or command prompt:

②

```
C:\>java -version
java version "1.7.0_11"
Java(TM) SE Runtime Environment (build 1.7.0_11-b21)
Java HotSpot(TM) 64-Bit Server VM (build 23.6-b04, mixed mode)
```

It is recommended that your computer has Java version 1.6 or later and your computer has at least 2 gigabytes of RAM. If you do not have Java or you need to update to a more recent version, you can do so by clicking the Java icon on the right:

Java

Download Latest Release (Rockhopper version 2.0.3)

Rockhopper for Windows	Rockhopper for Mac	Rockhopper for any platform	Rockhopper source code
	Error opening Rockhopper on Mac?	To execute the GUI version of Rockhopper, use the following command: `java -Xmx1200m -jar Rockhopper.jar` To execute the command line version of Rockhopper, use the following command: `java -Xmx1200m -cp Rockhopper.jar Rockhopper`	To extract the source code from the JAR file, use the following command: `jar xf Rockhopper.jar` To extract the source code from the compressed TAR archive, use the following command: `tar xjf Rockhopper-2.0.3.tar.bz2`

①Javaのバージョンを確認。
②赤枠部分が次のスライド

W2-2：Java確認など

De novo transcripton ssembly (Rockhopper2 ver. 2.0.3)

- Rockhopper2(バクテリア用): Tjaden B, Genome Biol., 2015
- Java確認など[W2-2]

```
cd

pwd
java -version
cd ~/Downloads
pwd
ls
```

- wgetで取得[W2-3]

```
wget -c http://cs.wellesley.edu/~btjaden/Rockhopper/download/current/Rockhopper.jar
ls
ls -l Rockhopper.jar
```

- GUI版を実行[W2-4]

```
ls -l Rockhopper.jar
java -Xmx1200m -jar Rockhopper.jar
```

W2-2：Java確認など

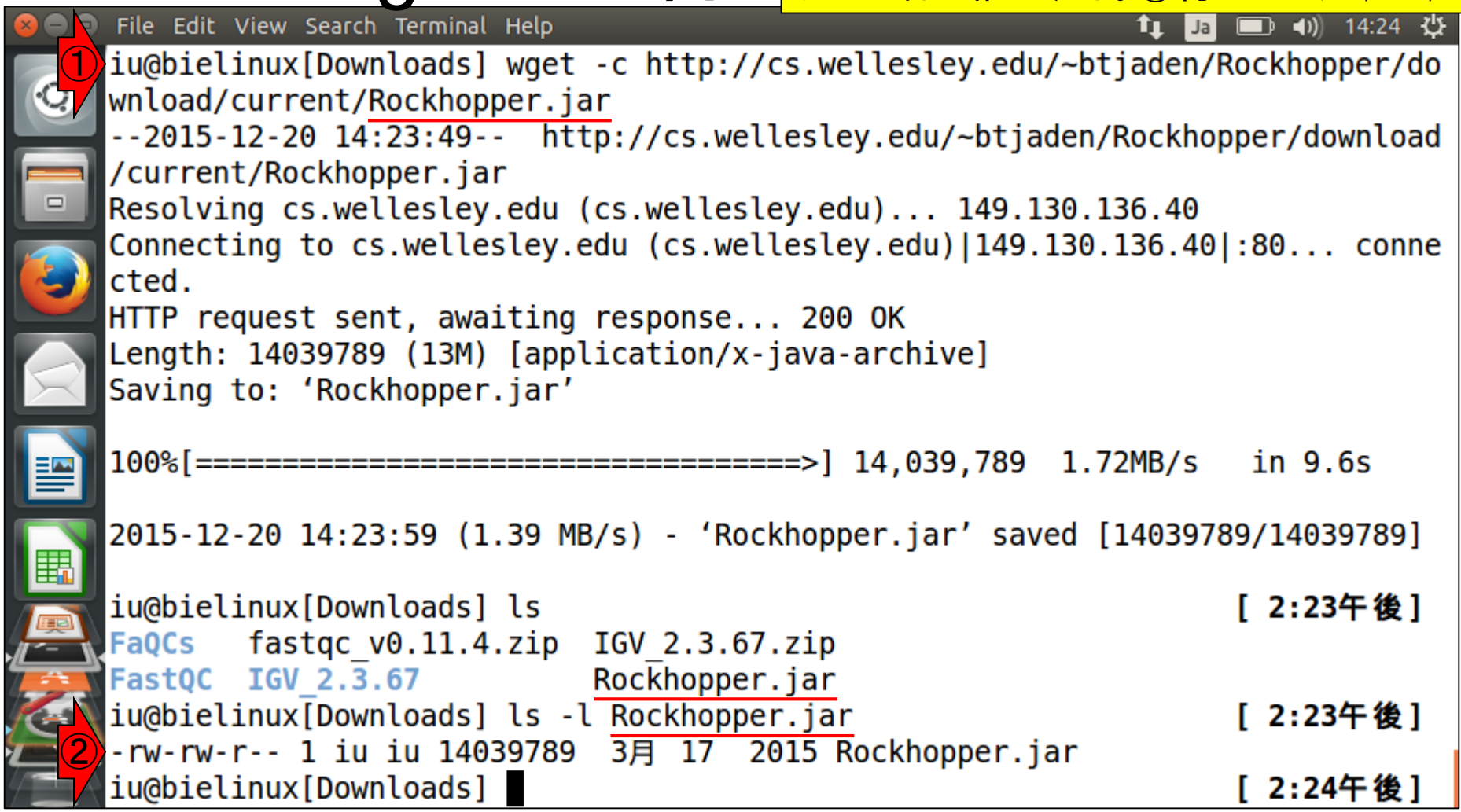
①「java -version」実行結果（作業ディレクトリはどこでもよい）。このPCには、ver. 1.7.0_79がインストールされていることがわかる。②~/Downloadsに移動。③ls実行結果で見えるものはヒトによって異なるが、基本気にしなくてもよい。④講習会では、ダウンロード済みなのでRockhopper.jarが見えている

```
iu@bielinux[~/Downloads]
iu@bielinux[iu] pwd
/home/iu
①iu@bielinux[iu] java -version
java version "1.7.0_79"
OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
②iu@bielinux[iu] cd ~/Downloads                                    [ 2:07午後]
iu@bielinux[Downloads] pwd                                         [ 2:07午後]
/home/iu/Downloads
③iu@bielinux[Downloads] ls                                         [ 2:07午後]
boost_1_61_0.tar.bz2          kmergenie-1.6982.tar.gz
Bridger_r2014-12-01.tar.gz    master.zip
FaQCs                         Rockhopper.jar
FastQC                        sratoolkit.2.6.3-ubuntu64.tar.gz
fastqc_v0.11.4.zip            v2.2.0.tar.gz
IGV_2.3.67                    velvet_1.2.10.tgz
IGV_2.3.67.zip
iu@bielinux[Downloads]                                             [ 2:07午後]
```

④

W2-3：wgetで取得

やらずに見るだけ!①wget実行したときのイメージ。赤下線部のURL情報の最後がダウンロードしたいファイル名に相当する。②約13MB (14,039,789 bytes)



W2-4：GUI版を実行

①W2-4。Rockhopper2のGUI版を実行。②赤枠部分が次のスライド。以後はこのようなウェブページのスクリーンショットを基本的に示さないので、自分で辿っていってください

- wgetで取得[W2-3]

```
wget -c http://cs.wellesley.edu/~btjaden/Rockhopper/download/current/Rockhopper.jar
ls
ls -l Rockhopper.jar
```

- GUI版を実行[W2-4]

```
ls -l Rockhopper.jar
java -Xmx1200m -jar Rockhopper.jar
```

- Tips: バックグラウンドジョブ(background job)[W3-1]

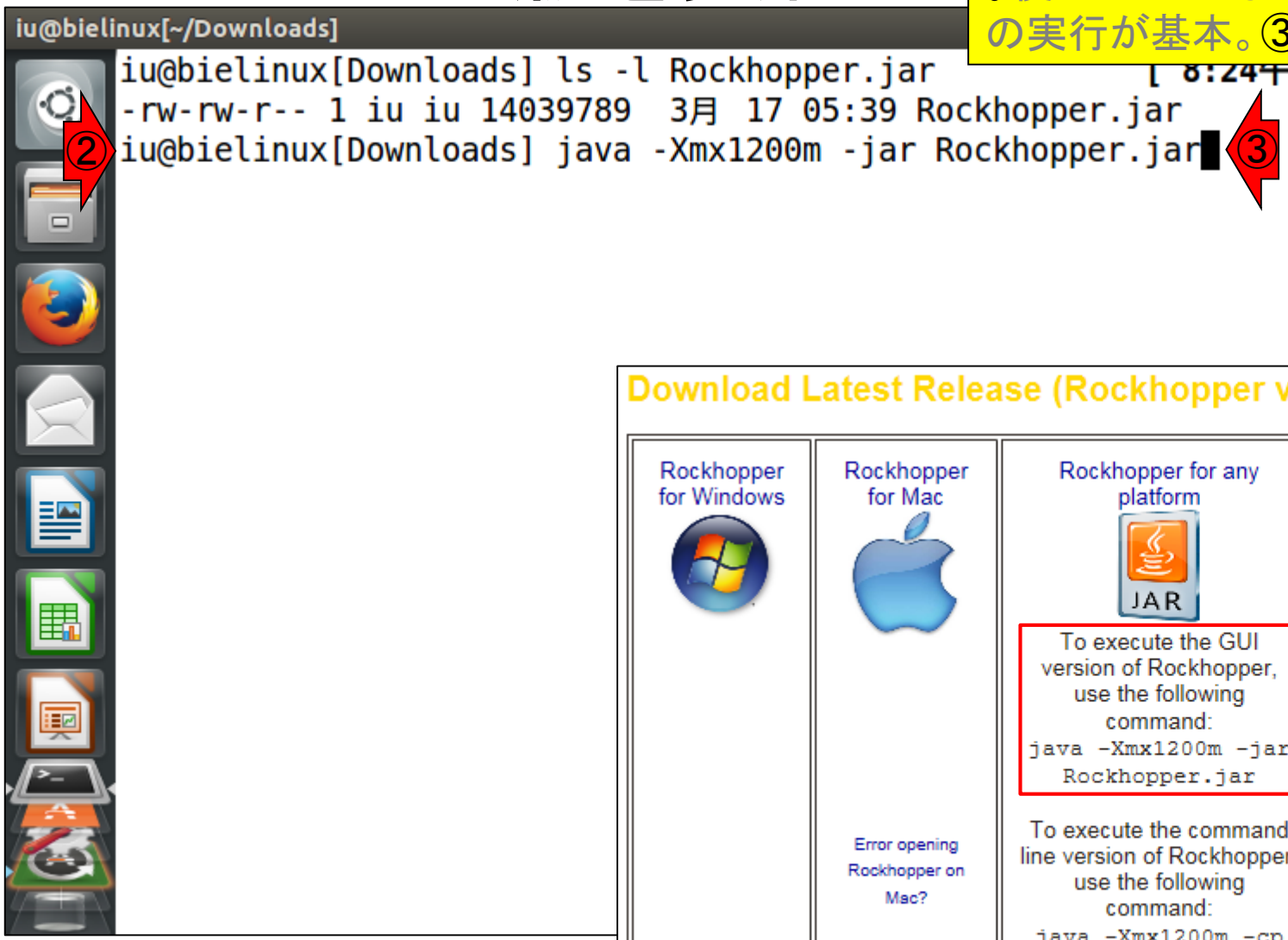
```
java -Xmx1200m -jar Rockhopper.jar&
```

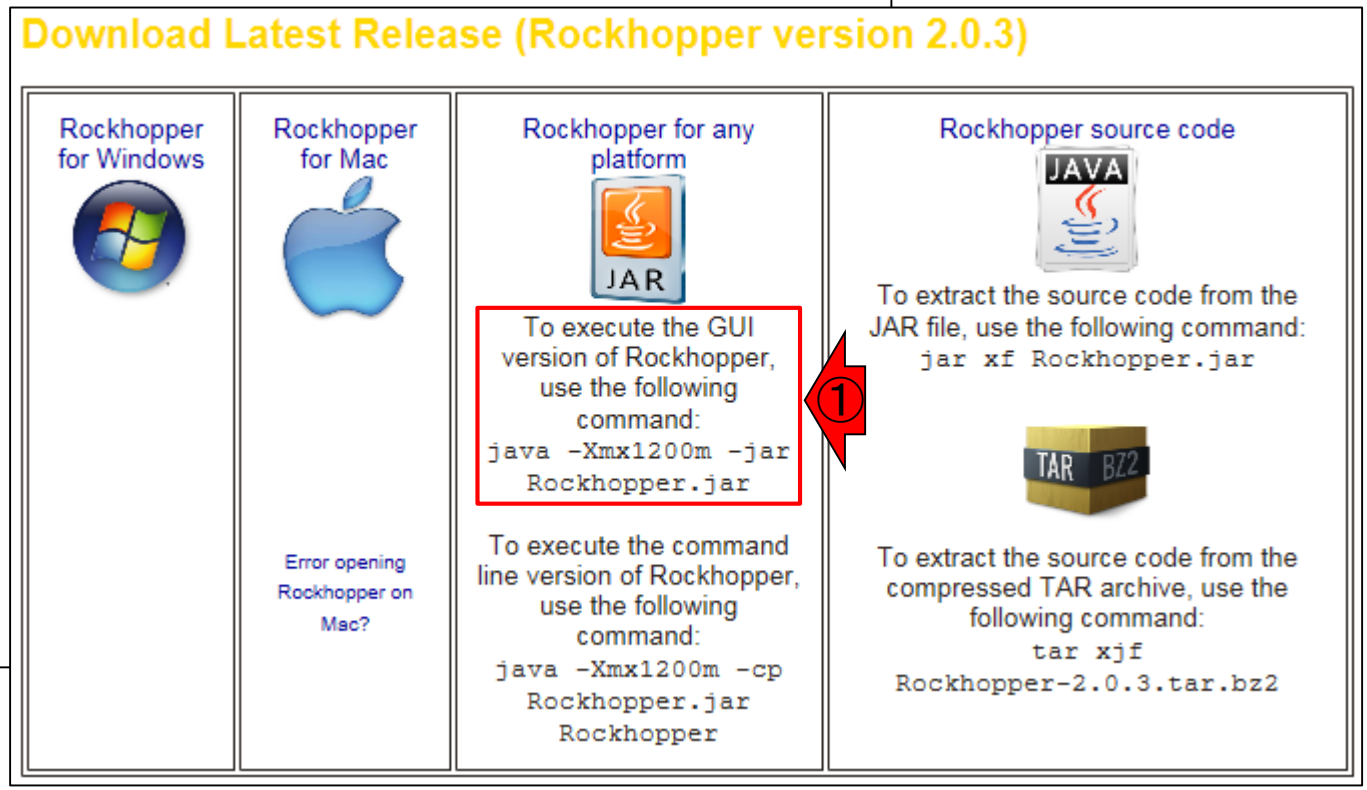
- Tips: psコマンド[W3-3からW3-5]

```
ps -f

ps -f
```

W2-4：GUI版を実行

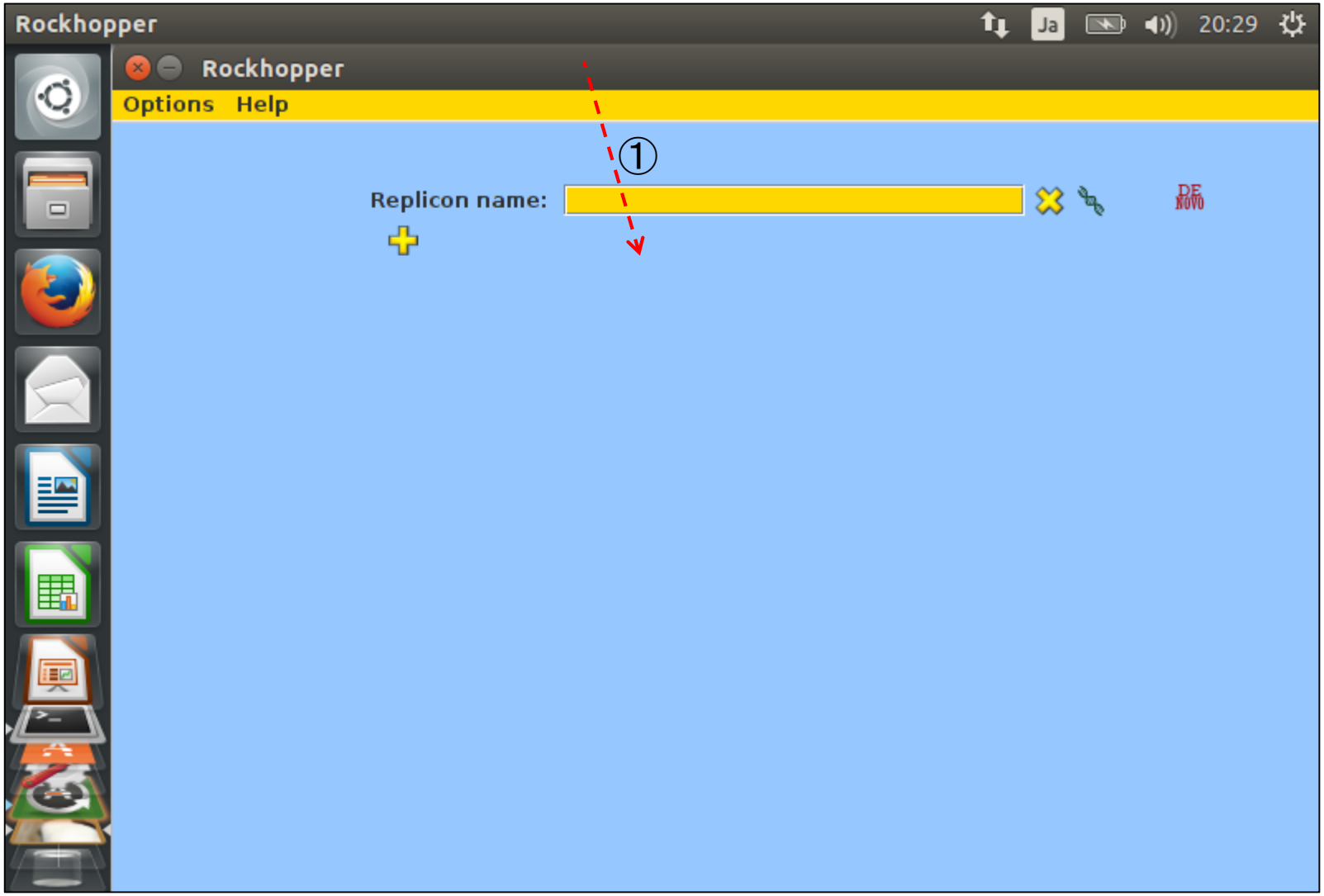
①RockhopperのGUI版を実行したい場合のやり方が書いてあるので、②その通りに実行。使ったことのないプログラムは、まずは例題の実行が基本。③リターンキーを押すと…





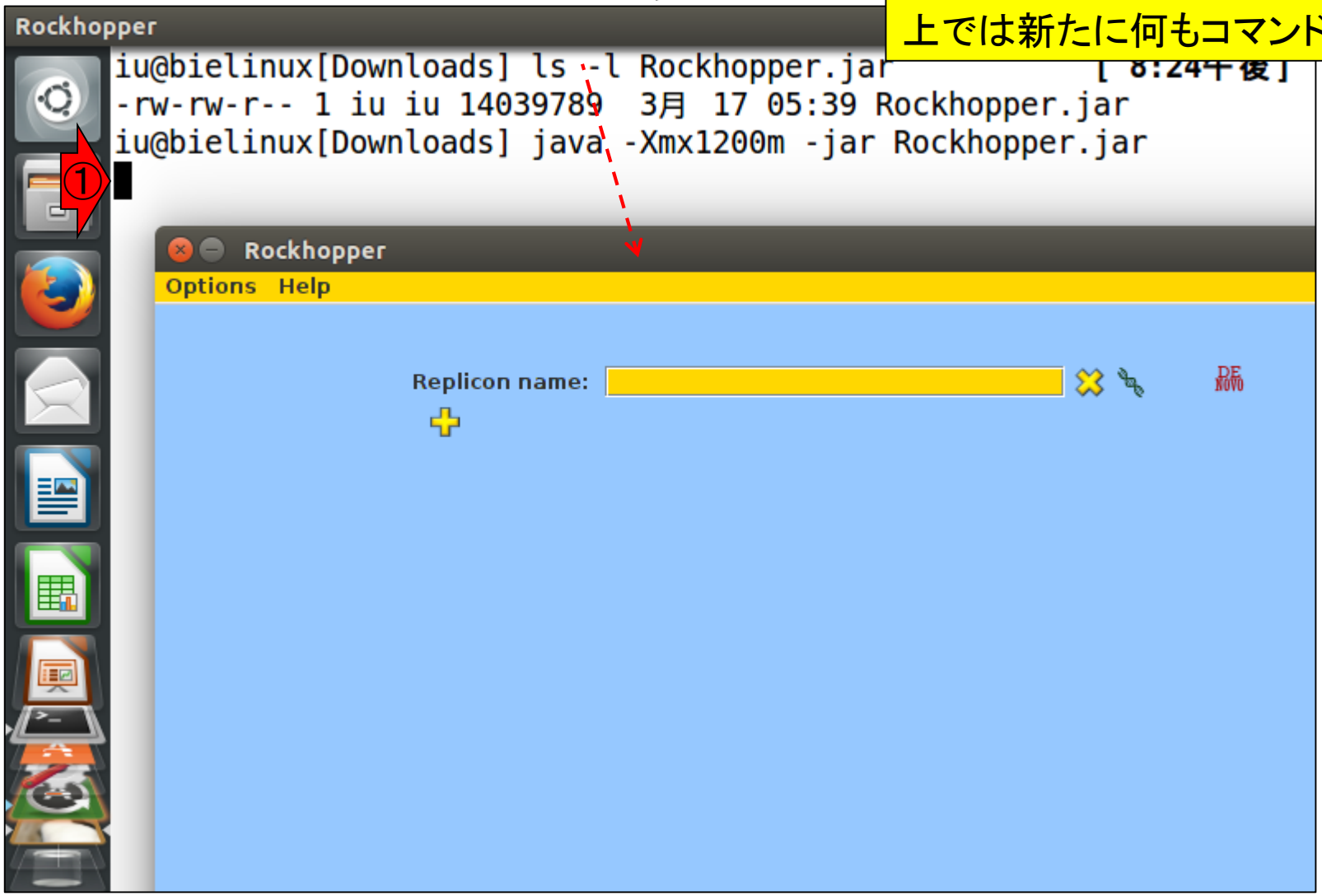
W2-4：GUI版を実行

RockhopperのGUI版が起動する。①GUIを赤矢印の始点から終点に移動させると…



W2-4：GUI版を実行

RockhopperのGUI版を起動したコマンドが見える。RockhopperのGUI起動中は、①のようにコマンドプロンプトが出ないため、このターミナル上では新たに何もコマンドを打つことができない

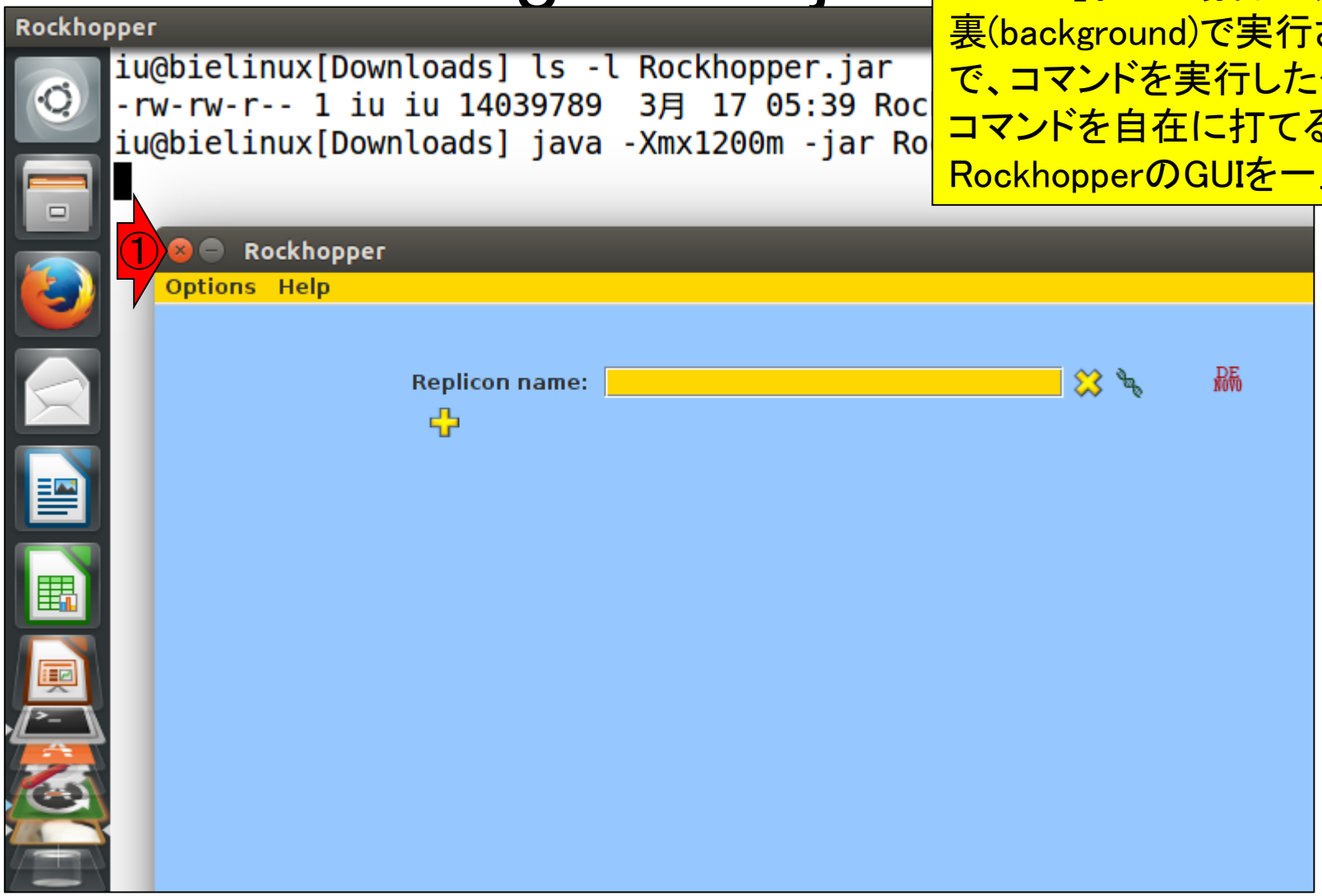


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

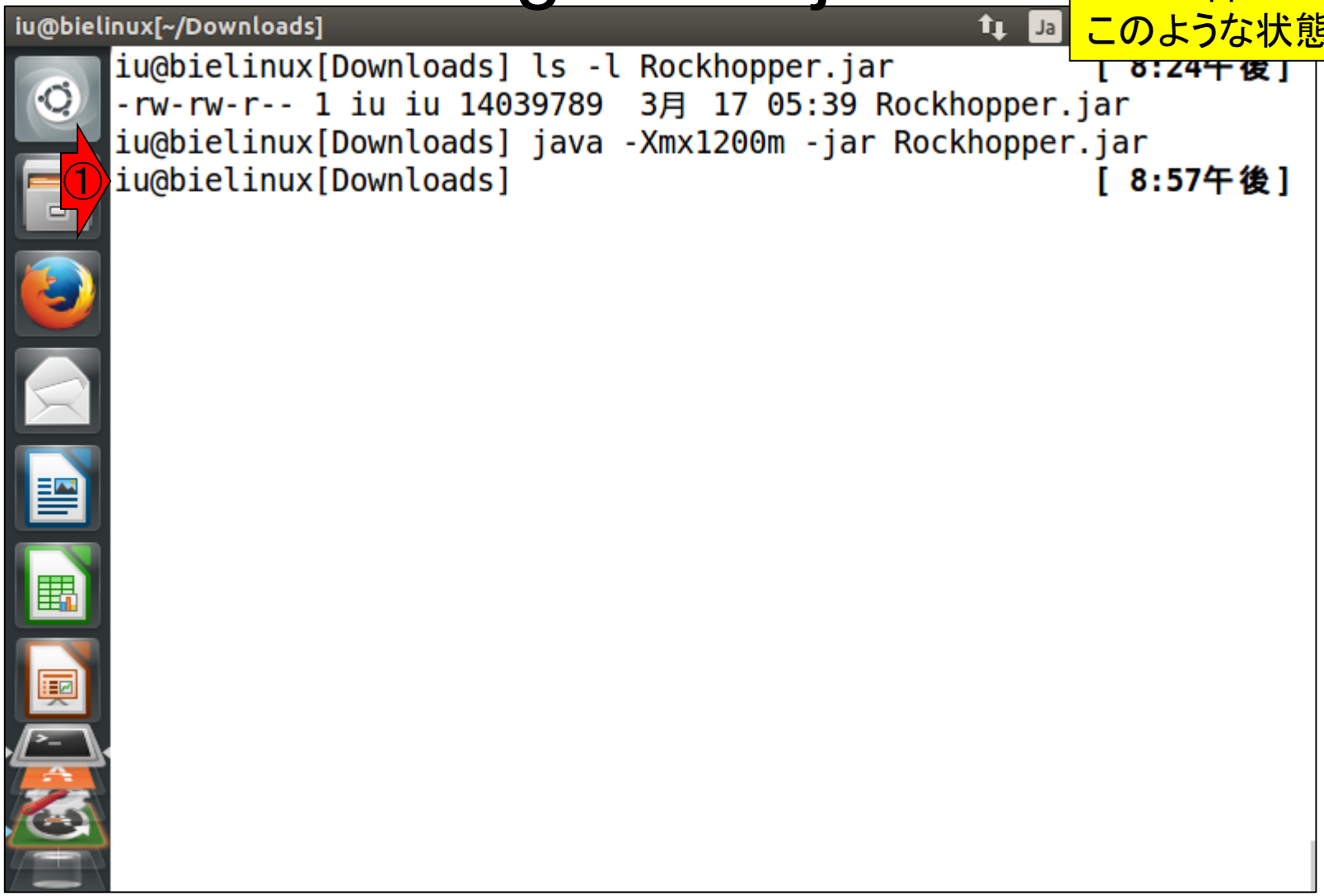
W3-1：background job

何もコマンドを打つことができない状態(W2-4)を回避する1つのやり方が「バックグラウンドジョブ」。この場合は、RockhopperのGUIを裏(background)で実行させる(jobを流す)ことで、コマンドを実行したターミナル上で、次のコマンドを自在に打てるようにすること。①RockhopperのGUIを一旦終了させると…



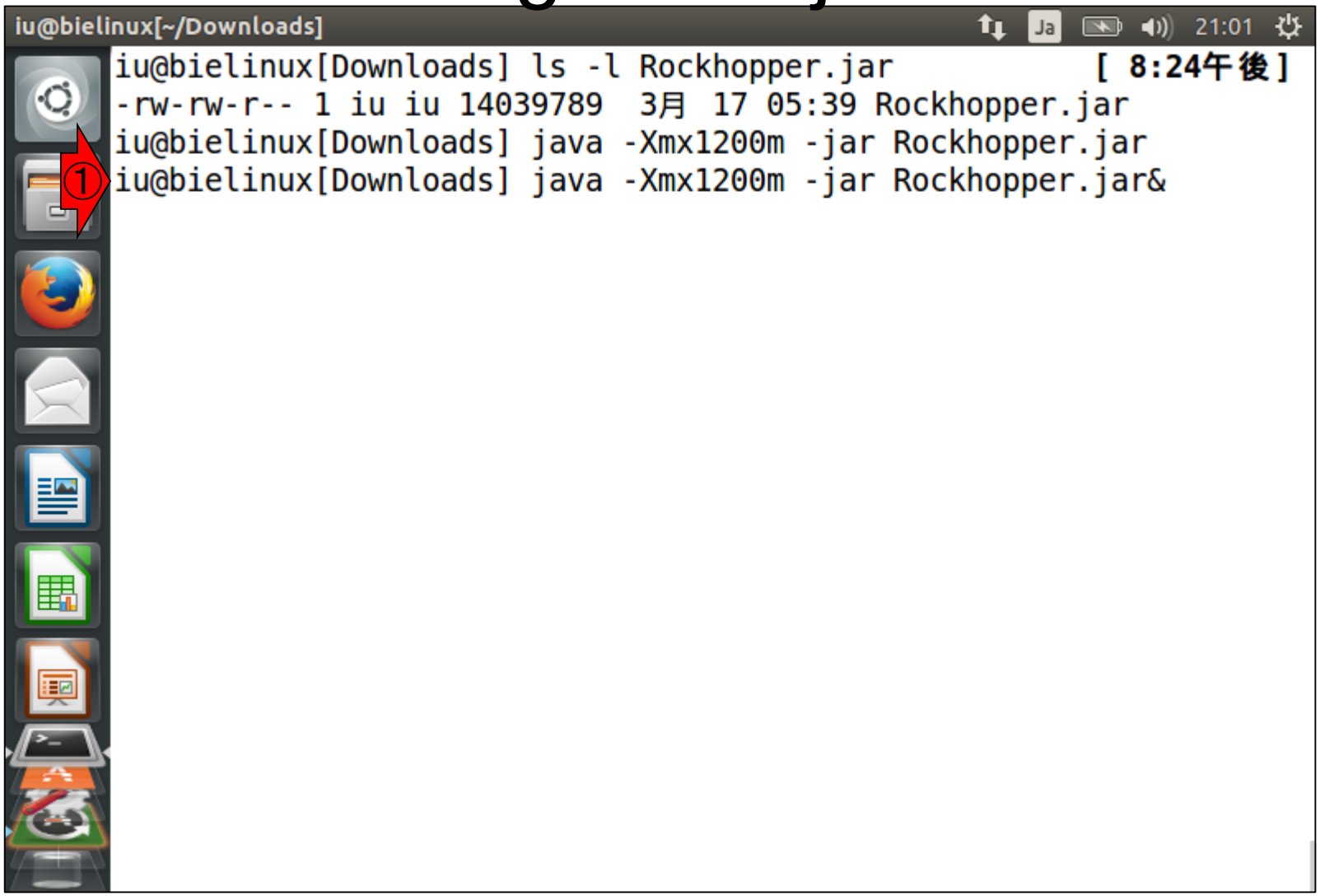
W3-1 : background job

①通常のコマンド打ち込み可能状態となる。バックグラウンドジョブとは、RockhopperのGUIを起動しつつも、このような状態にするテクニックです



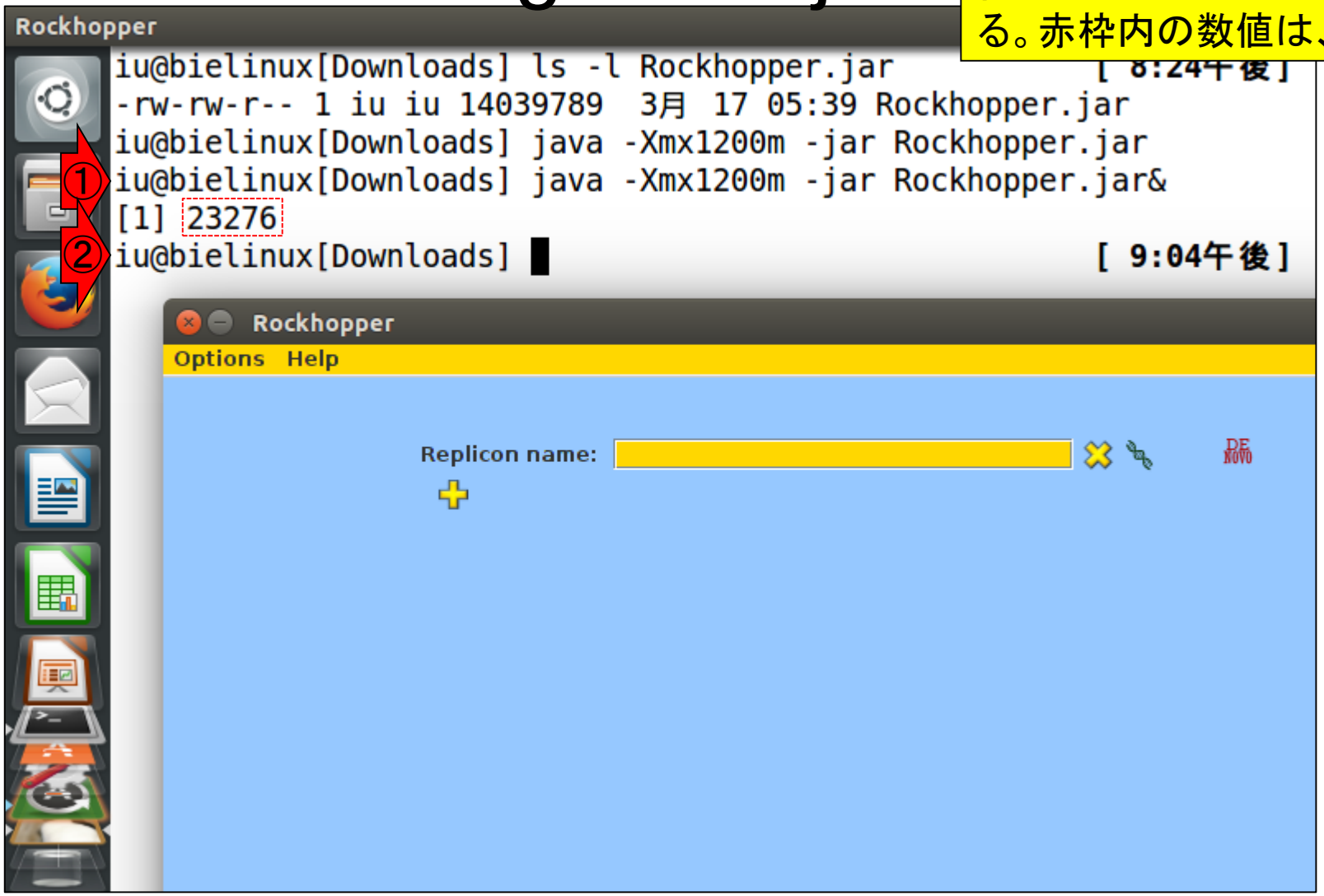
W3-1：background job

基本的には、①通常のコマンドの最後に「&」をつけるだけ



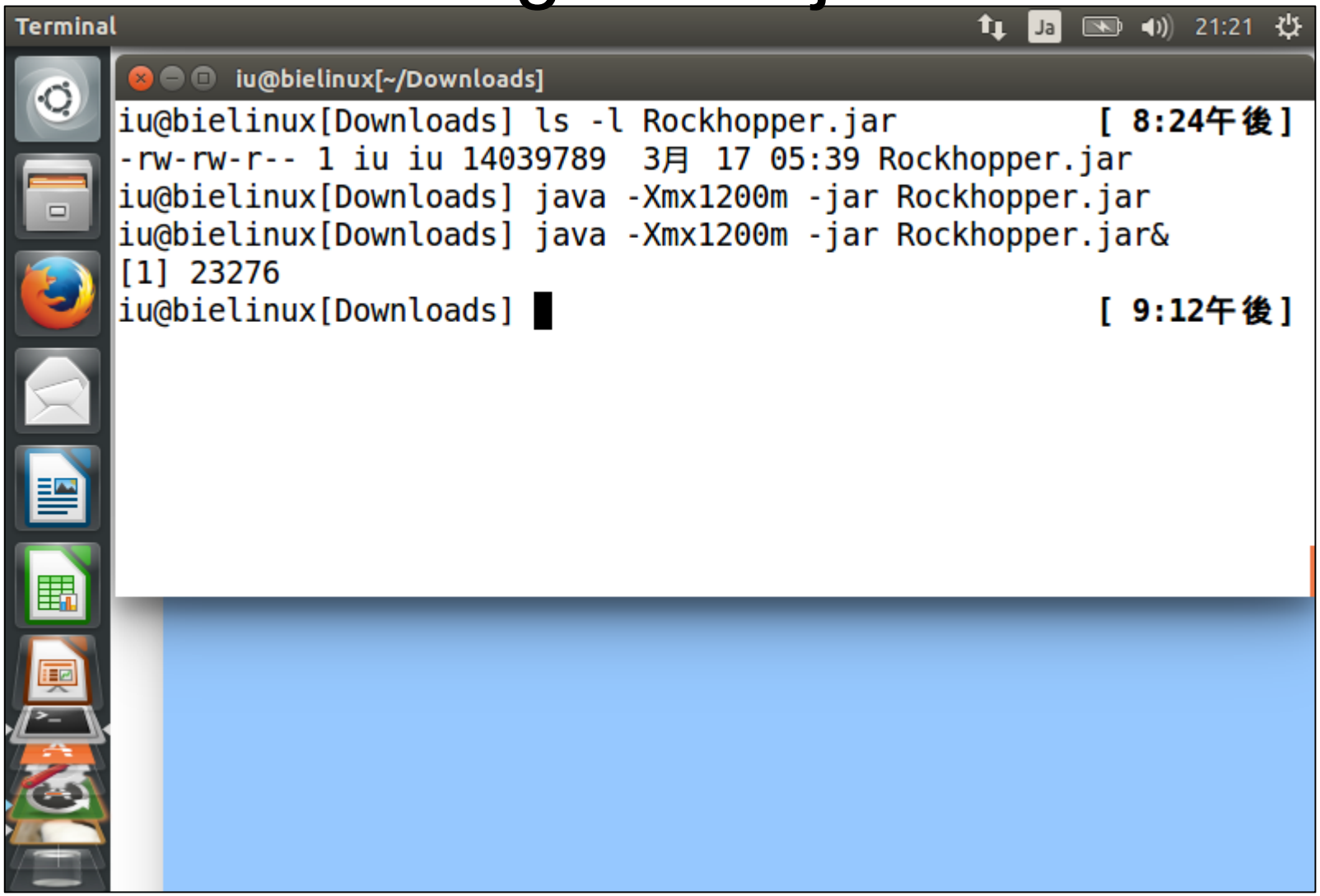
W3-2 : background job

①リターンキーを押した結果。確かにRockhopper GUIが起動しつつ、②コマンド打ち込み可能状態になっていることがわかる。赤枠内の数値は、ヒトによって異なる



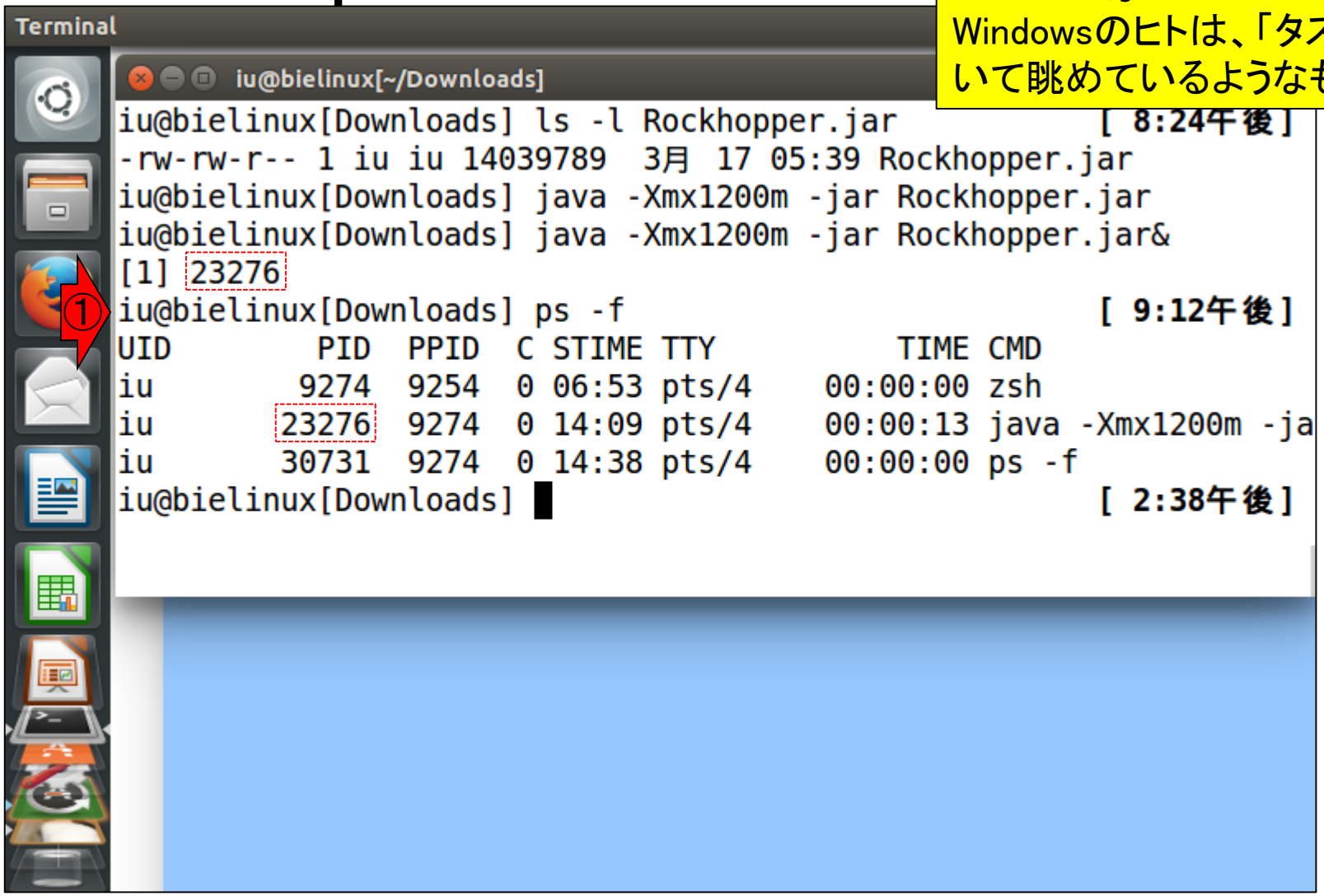
ターミナルの画面サイズを変更しただけ

W3-2 : background job



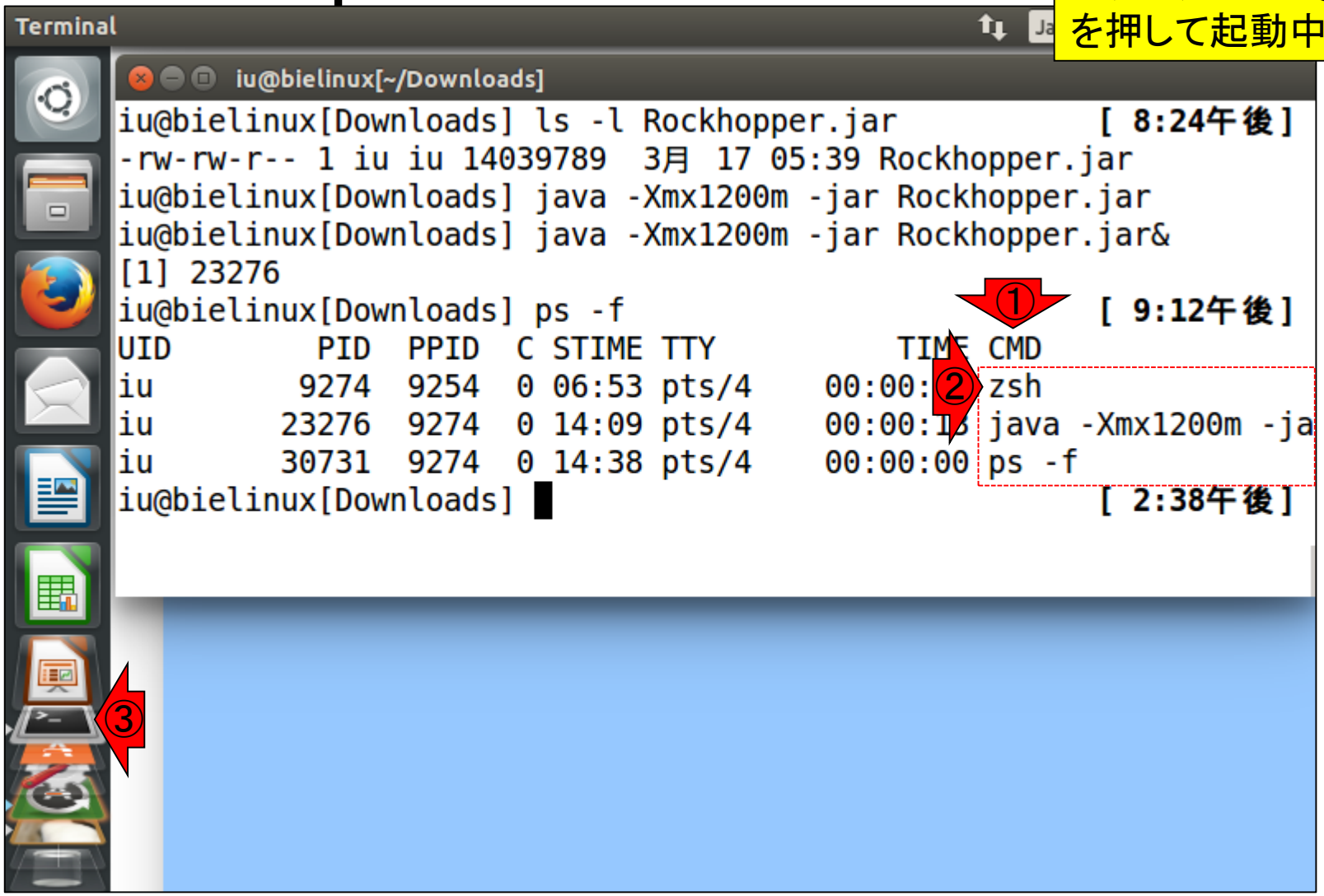
W3-3：psコマンド

①psコマンドで実行中のプロセスを表示。プロセスと表現する場合が多いのでそう書いているが、jobやタスクという理解でもよい。Windowsのヒトは、「タスクマネージャー」を開いて眺めているようなものだと思えばよい



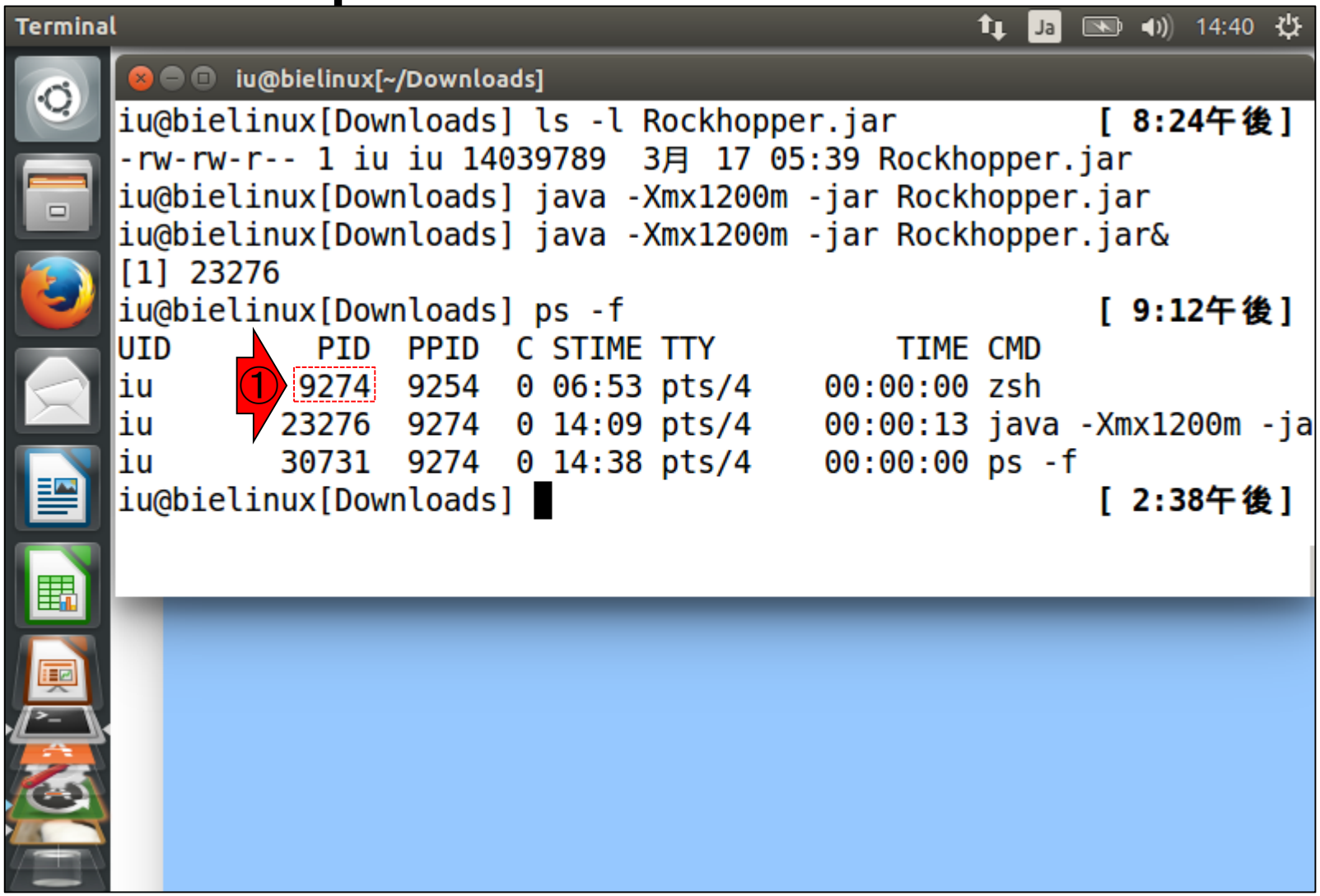
W3-3：psコマンド

①CMD列が、現在実行中のコマンド。②zshのみ打ち込んだ記憶がないだろうが、これは③のターミナルボタンを押して起動中のターミナルそのもの



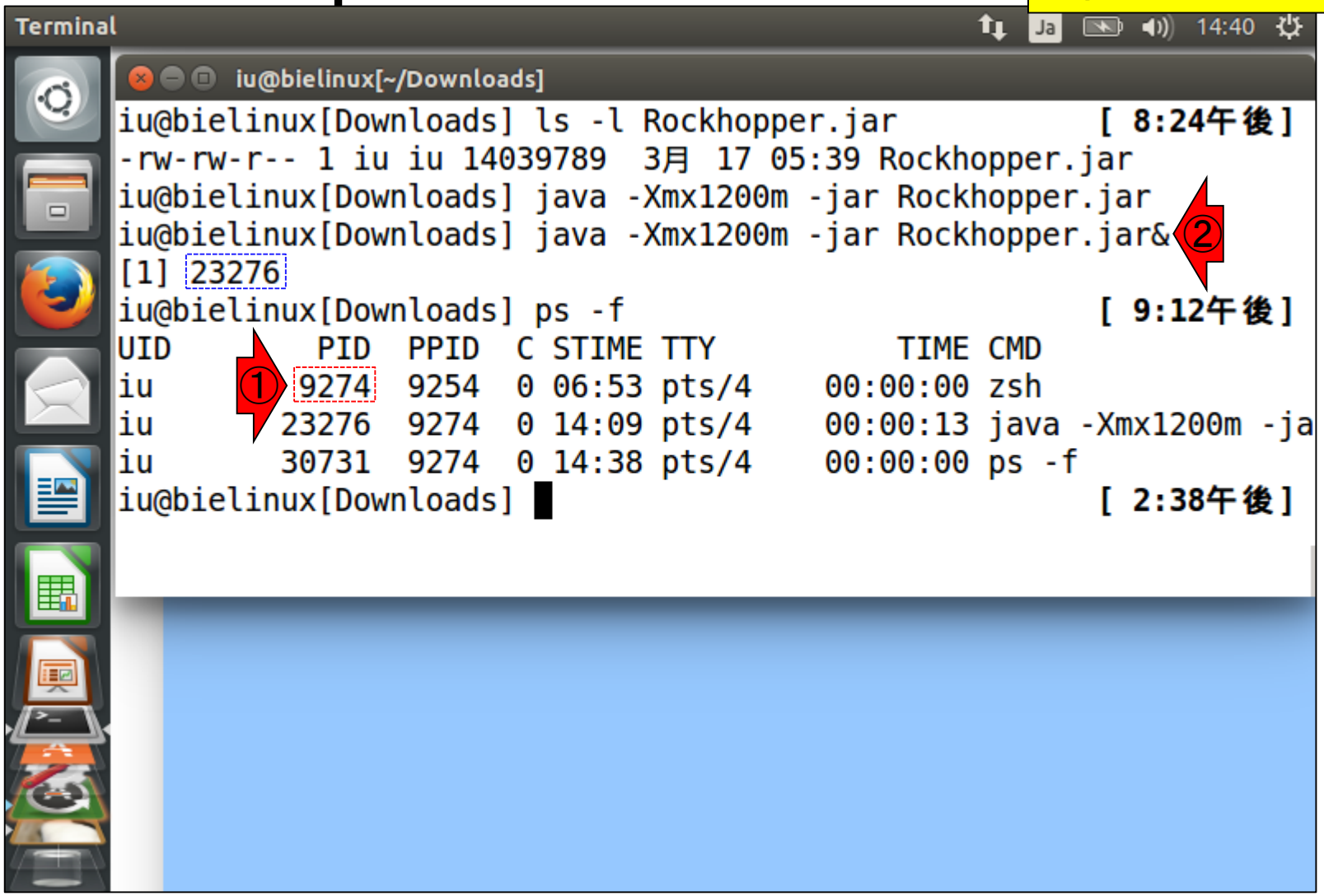
W3-4：psコマンド

①ターミナルボタンを押して起動したターミナルのプロセスID（PID)は9274



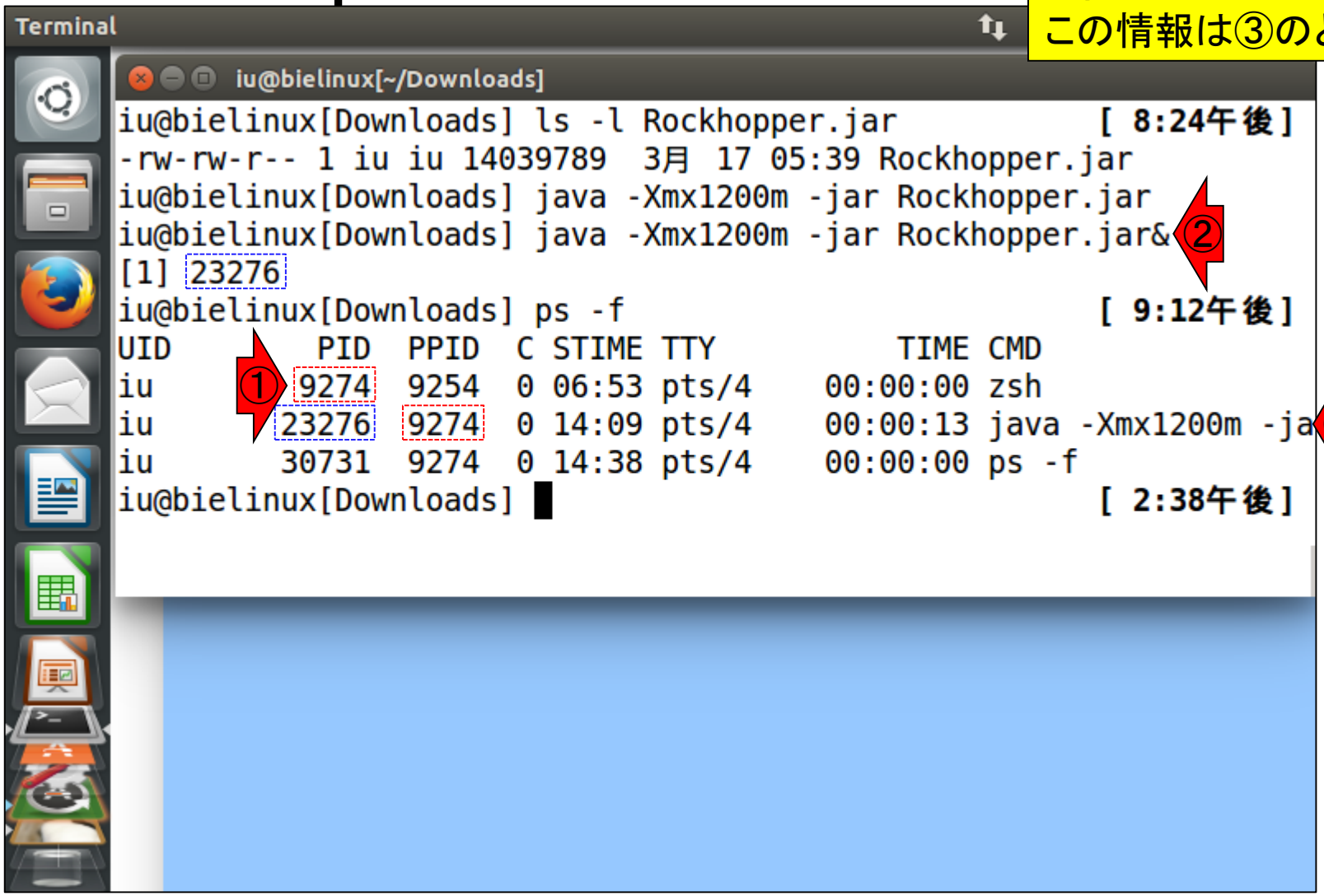
W3-4：psコマンド

①ターミナルボタンを押して起動したターミナルのプロセスID（PID)は9274。②で打ち込んだコマンドのPIDは23276



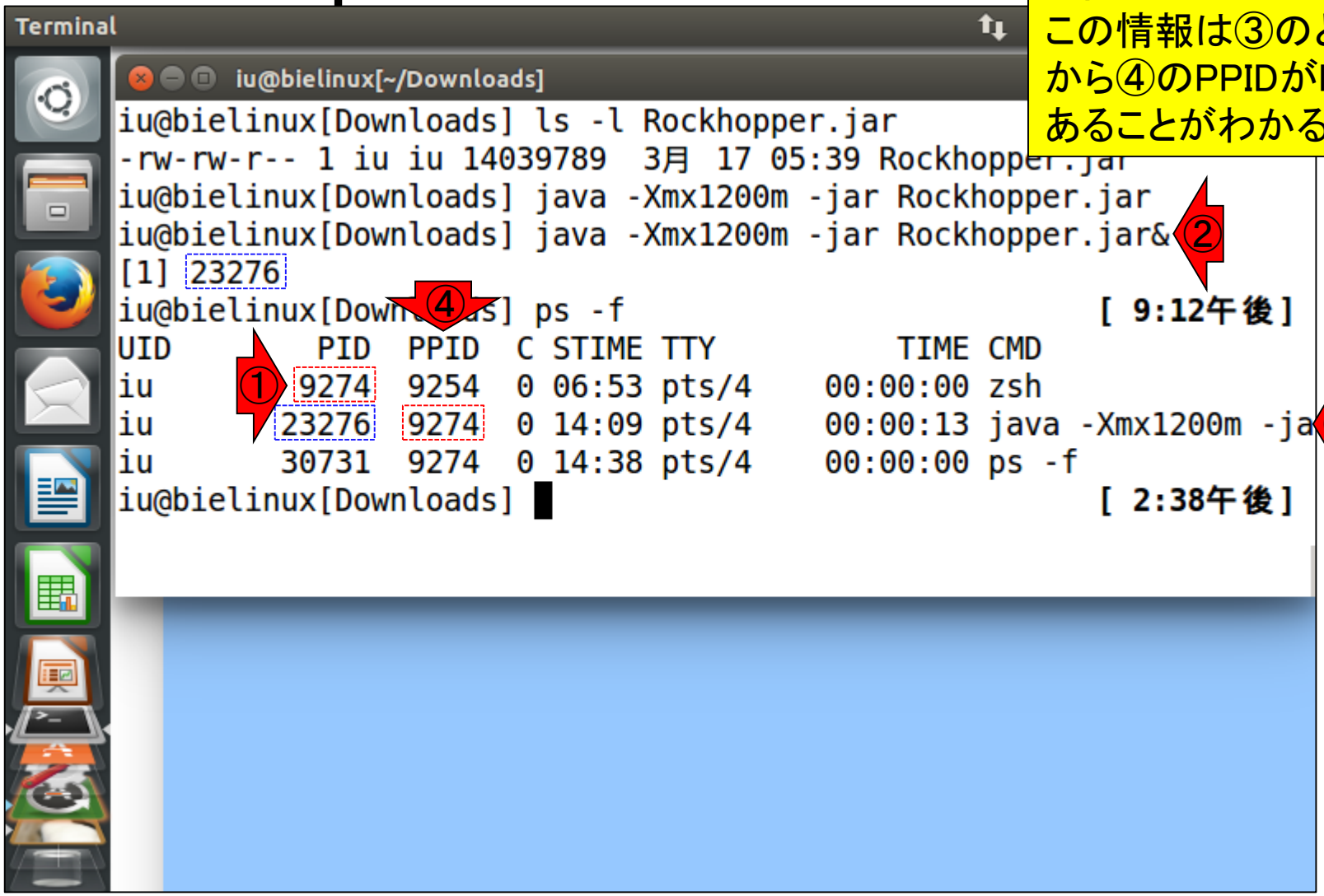
W3-4：psコマンド

①ターミナルボタンを押して起動したターミナルのプロセスID（PID)は9274。②で打ち込んだコマンドのPIDは23276。この情報は③のところに相当



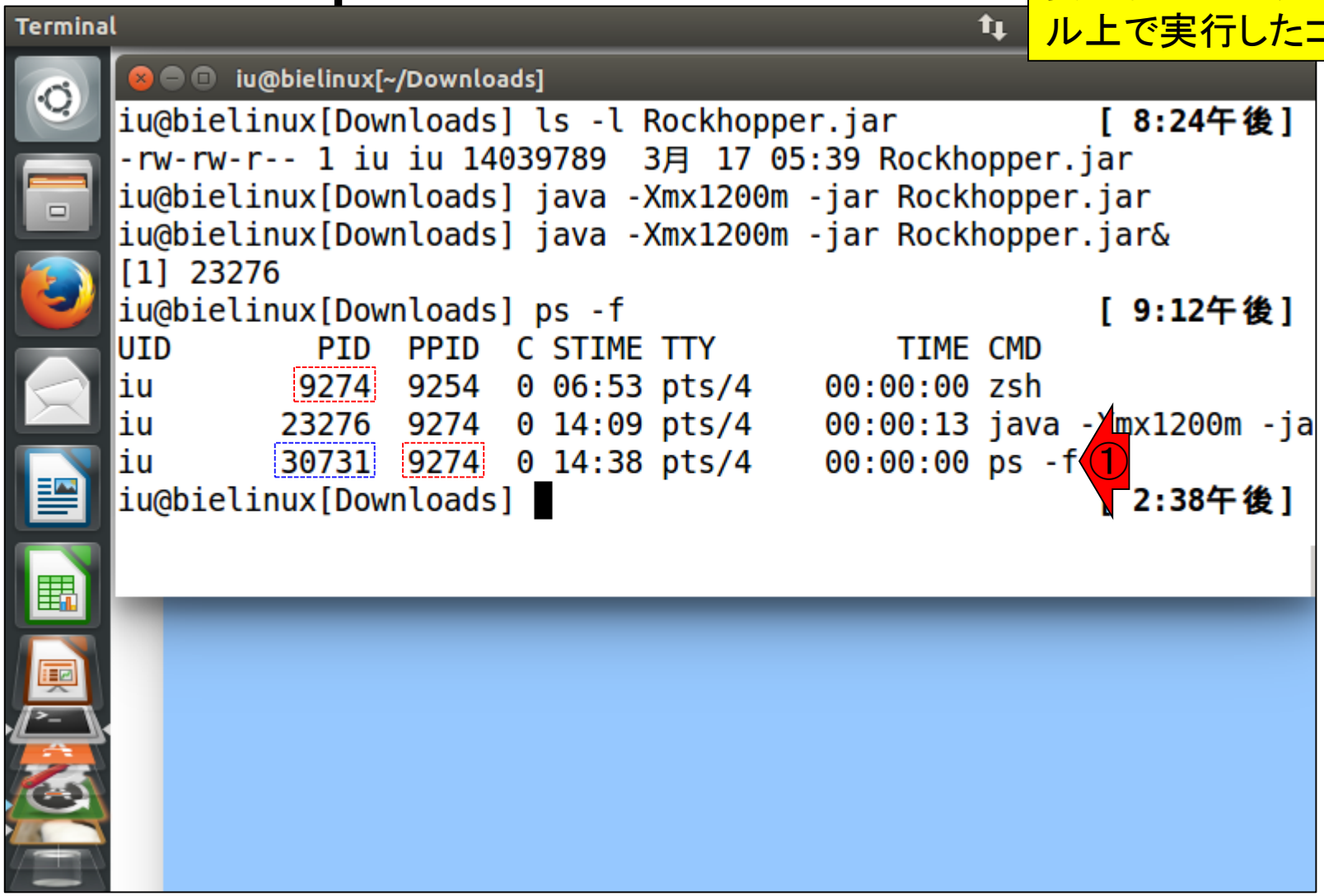
W3-4：psコマンド

①ターミナルボタンを押して起動したターミナルのプロセスID（PID)は9274。②で打ち込んだコマンドのPIDは23276。この情報は③のところに相当。全体像から④のPPIDがPIDの親プロセスIDであることがわかる。Parent PIDと解釈



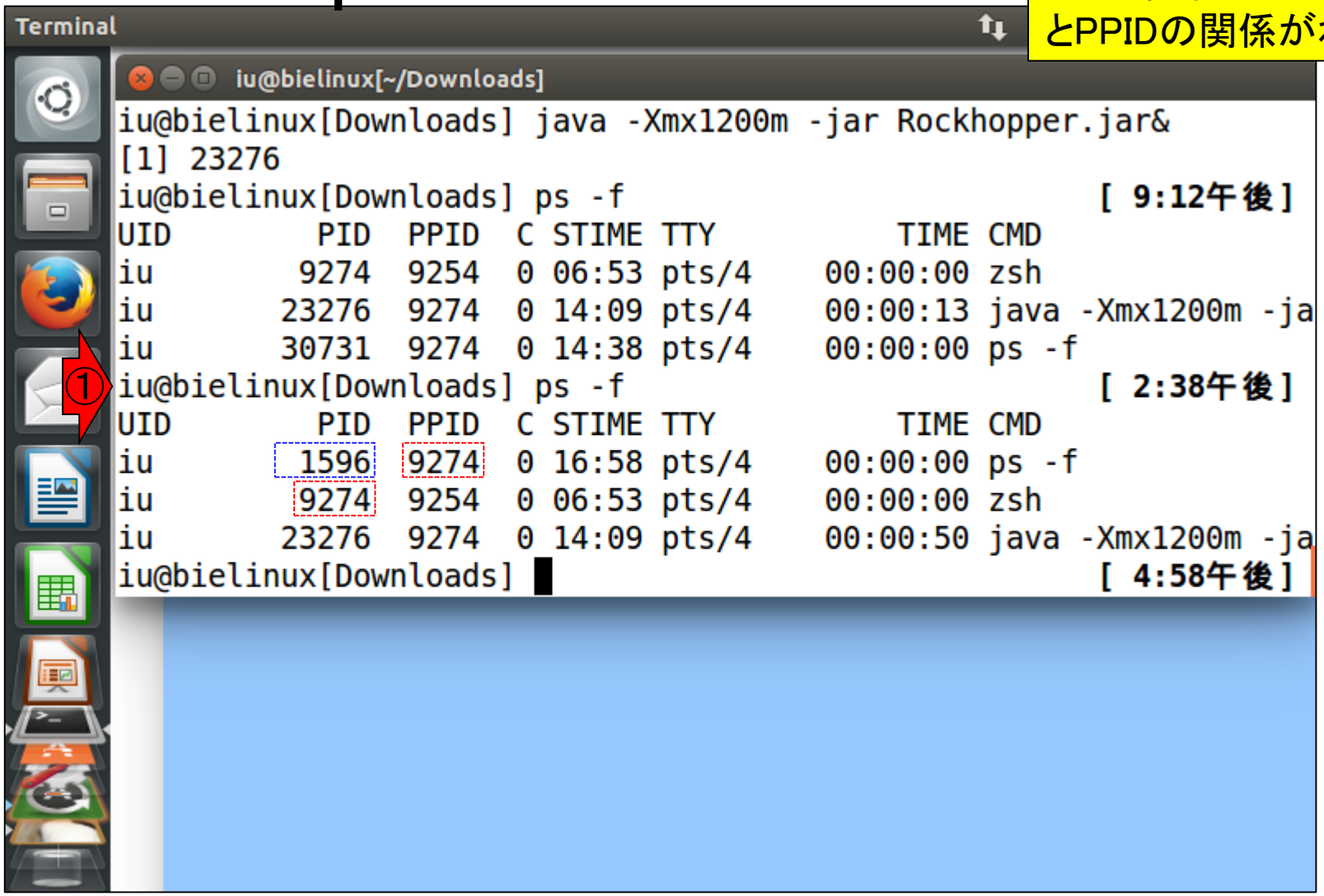
W3-5：psコマンド

①「ps -f」のプロセスID (PID)は30731。この親プロセスID (PPID)が9274なのは妥当。理由は、このPID9274のターミナル上で実行したコマンドだから



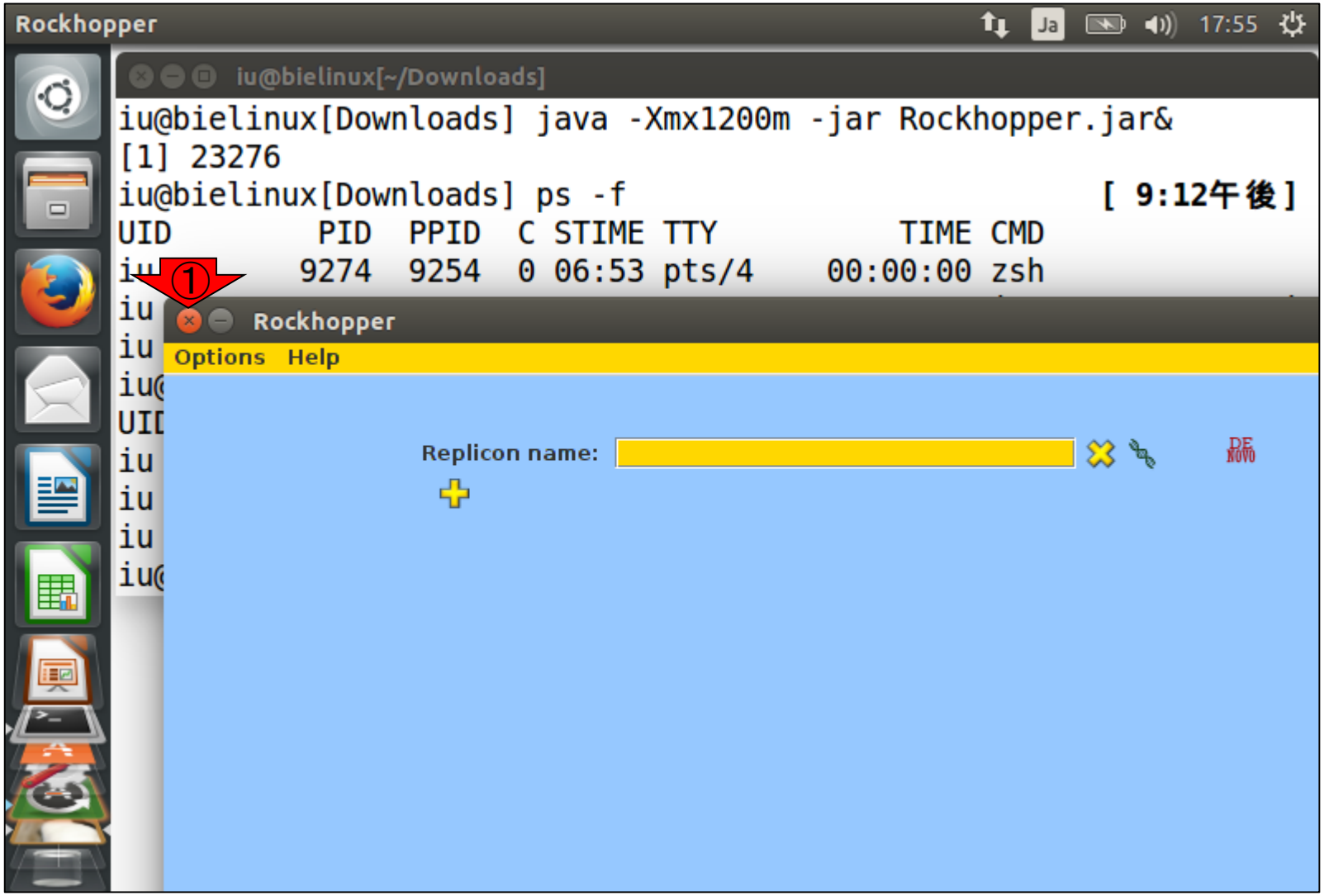
W3-5：psコマンド

①もう一度「ps –f」を実行。このプロセスID（PID）は1596。このように数値はコロコロ変わるものなので、基本的にPIDとPPIDの関係がわかっていればよい



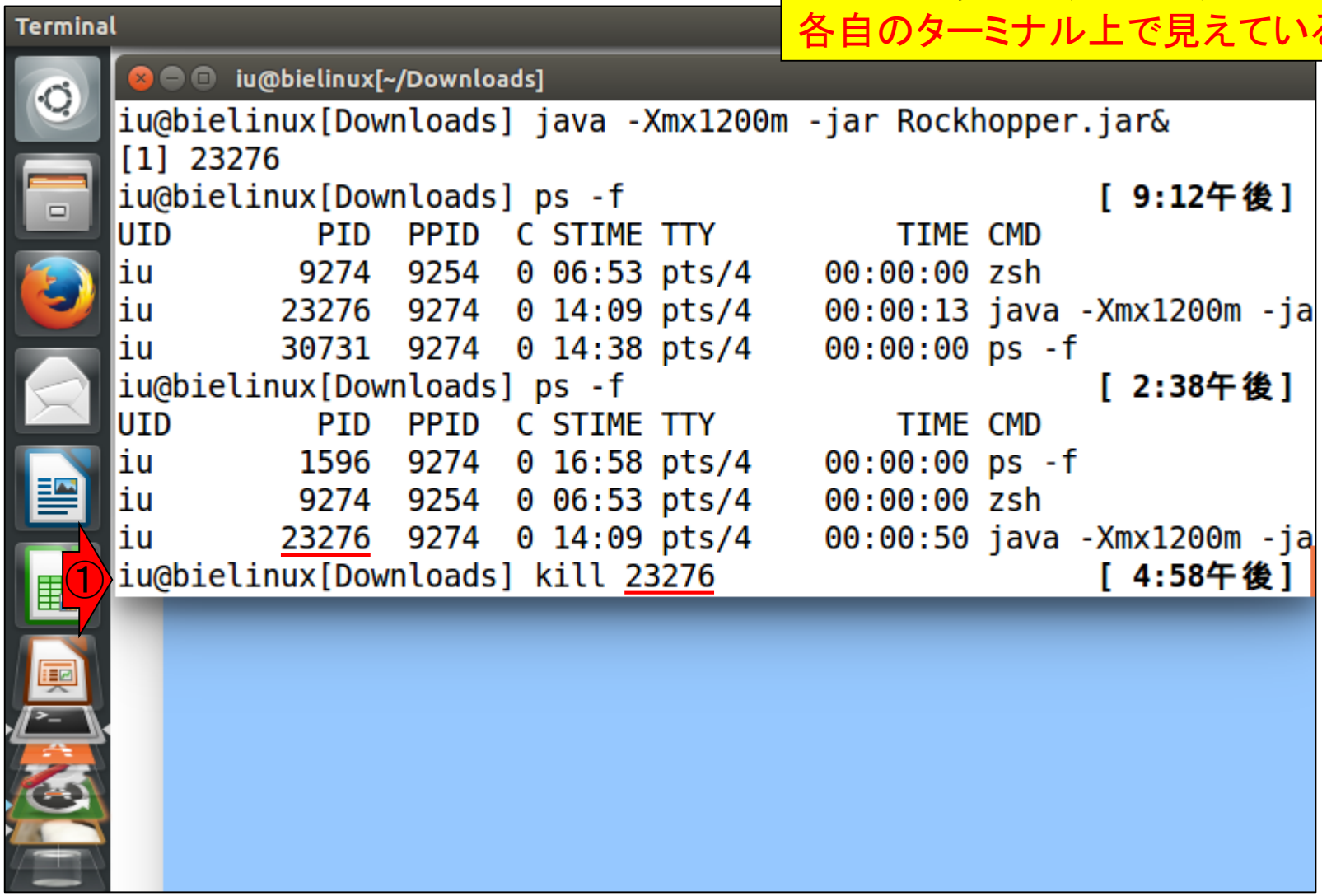
W3-6:プロセスの終了

①GUIベースでやる場合は、×ボタンだが、ここでは押さないで!



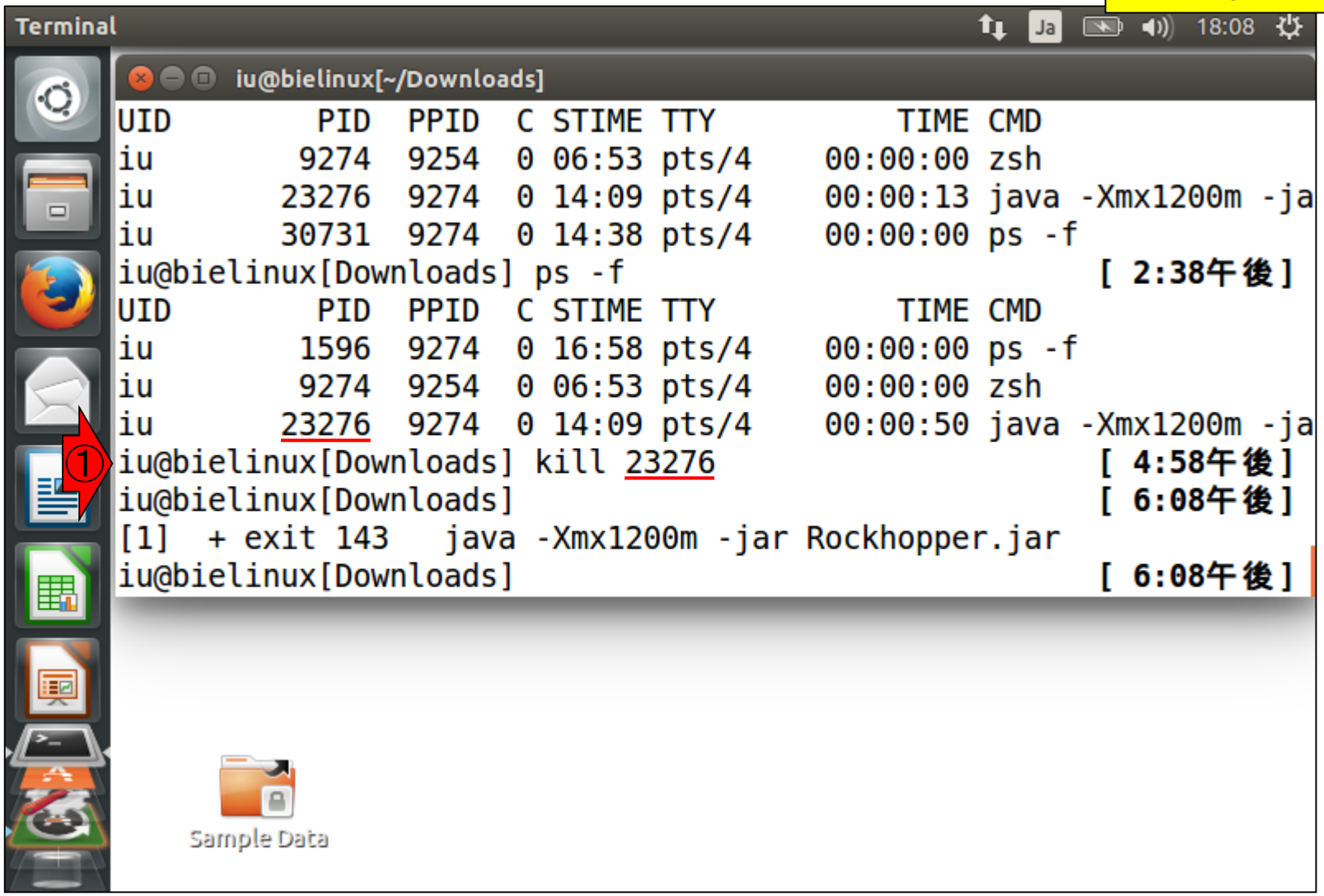
W3-6：killコマンド

①「kill *プロセスID*」で終了させることができる。やたらとメモリを消費している意味不明なプロセスが実行されている場合に、このような処理を行って終了させる。
各自のターミナル上で見えているPIDを与えてkillしよう



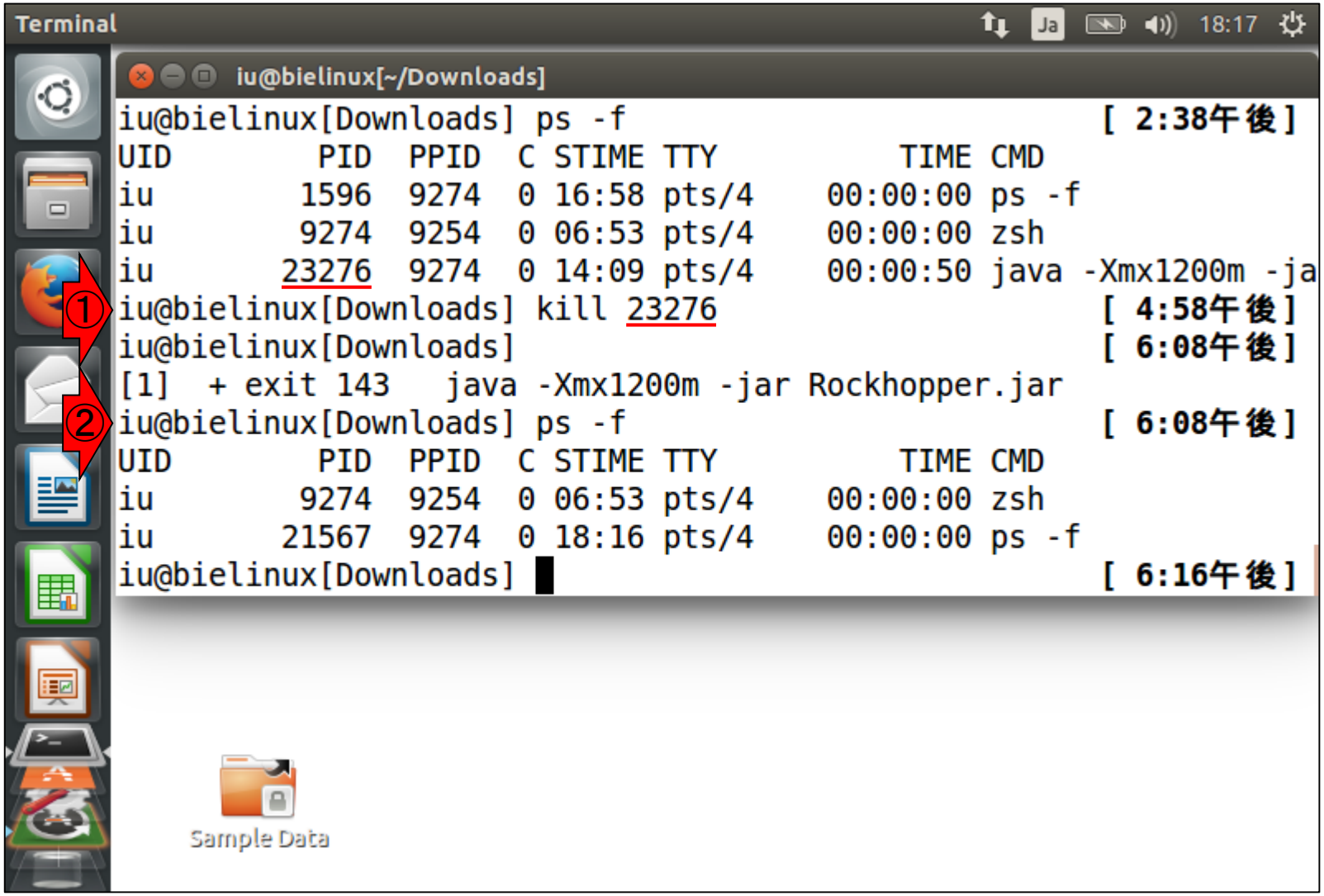
W3-6：killコマンド

①「kill *プロセスID*」でリターンキーを押した後の状態。Rockhopperの GUIが終了していることがわかる



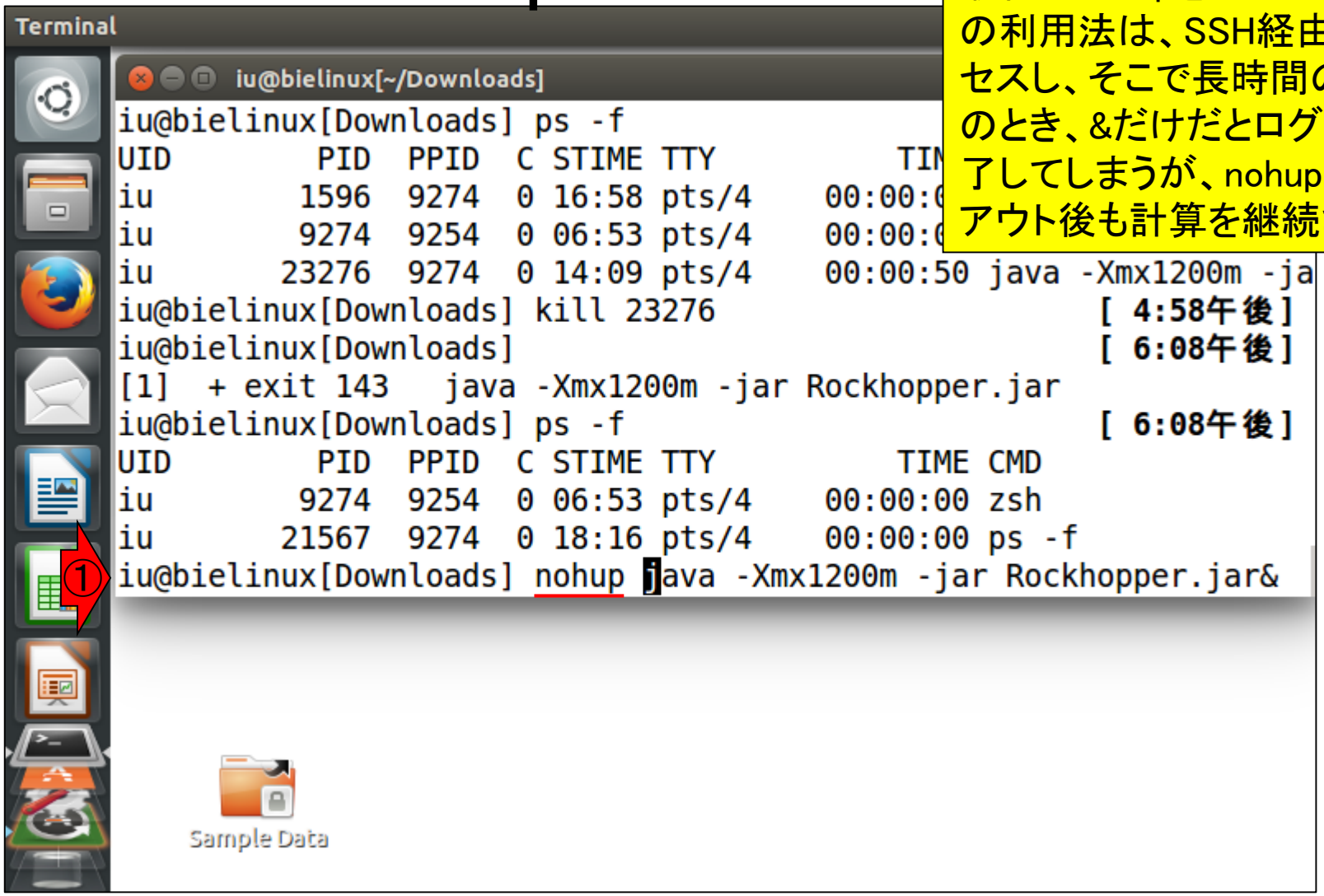
W3-6：killコマンド

②「ps −f」で確認。確かにPID23276は存在しない



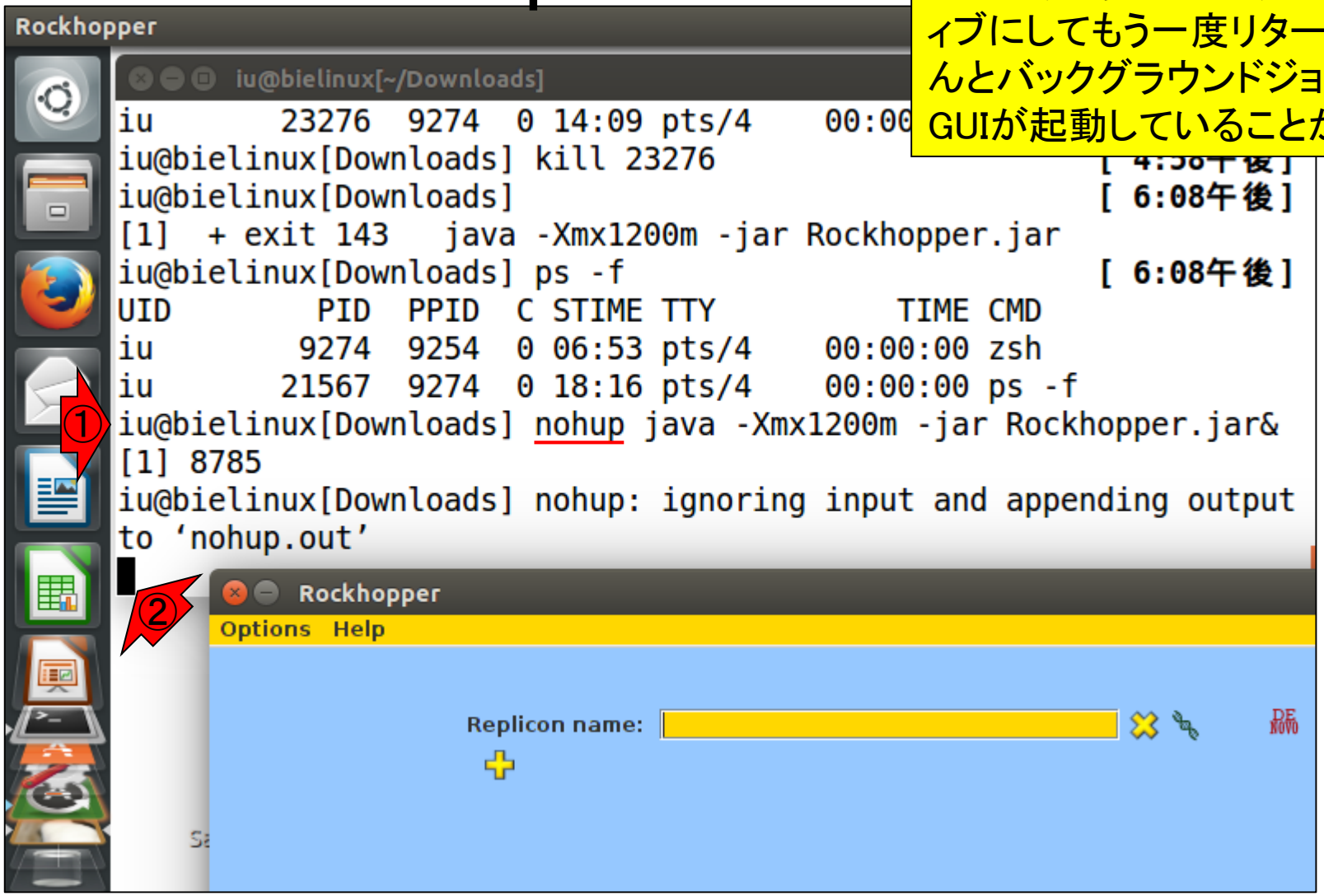
W3-7 : nohupコマンド

バックグラウンドジョブ時には、通常コマンドの最後に&をつけるだけでなく、①コマンドの最初にnohupをつける。一般的なNGS解析の利用法は、SSH経由で大型計算機にアクセスし、そこで長時間の計算を実行する。このとき、&だけだとログアウト時に計算が終了してしまうが、nohupをつけることで、ログアウト後も計算を継続させることができる



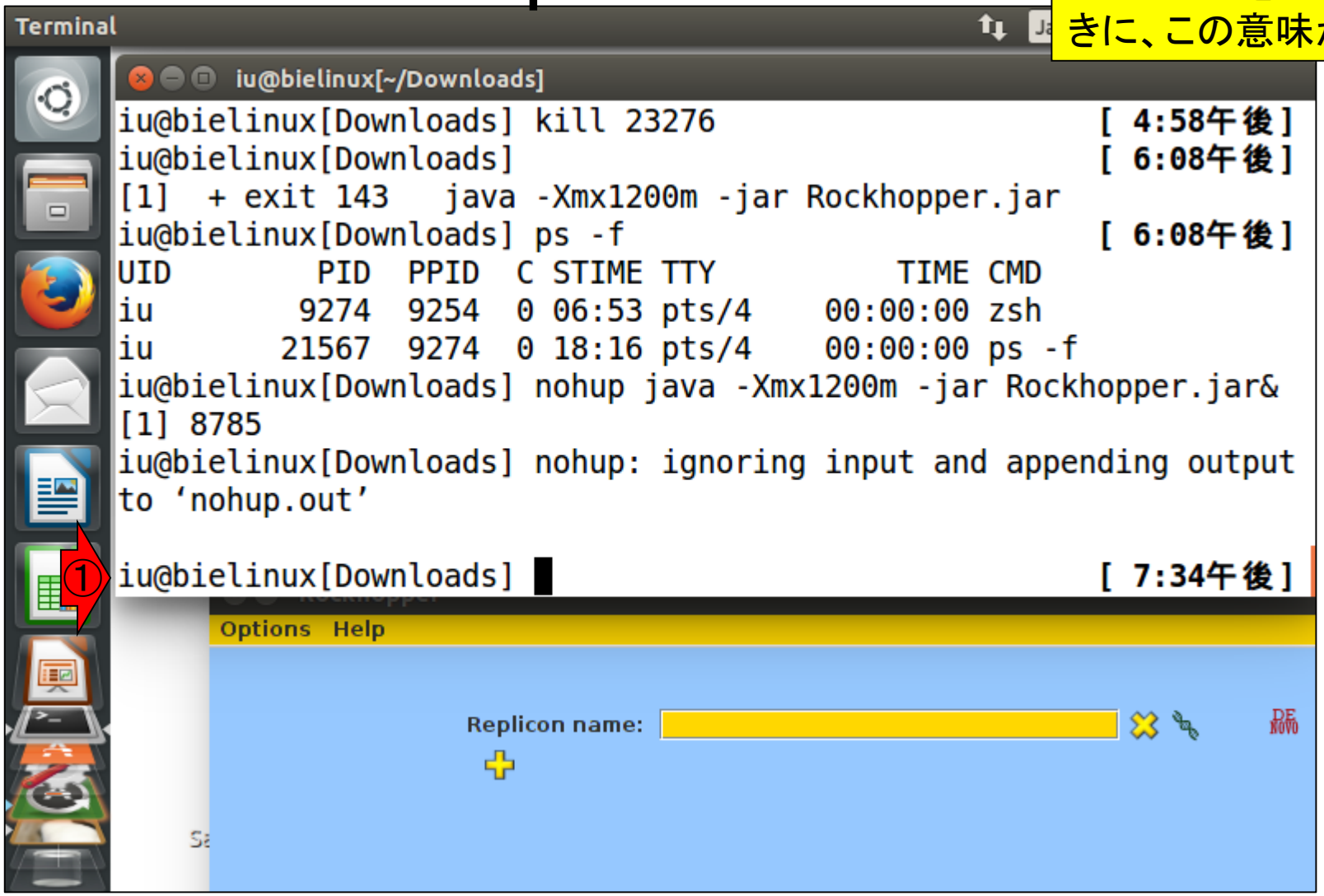
W3-7 : nohupコマンド

①でリターンキーを1回押した直後の状態。ターミナル上では、一見コマンド打ち込み不可能なように見えるが、ターミナル画面をアクティブにしてもう一度リターンキーを押すと、ちゃんとバックグラウンドジョブとしてRockhopper GUIが起動していることが確認できる。



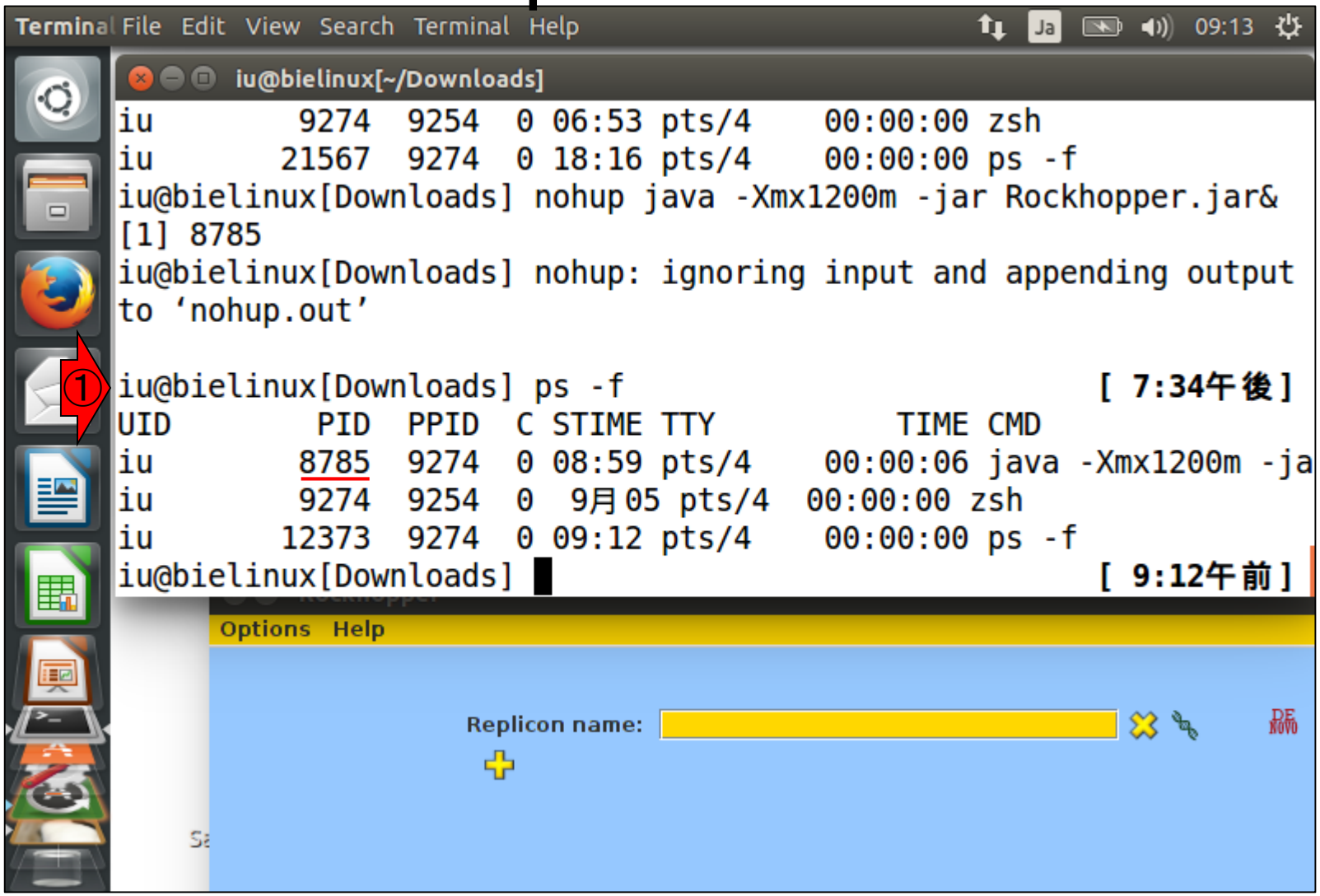
W3-7 : nohupコマンド

①SSHの話までするとややこしいのでこれ以上深入りしないが、遺伝研スパコンなどを利用するようになったときに、この意味が理解できるでしょう



W3-7：nohupコマンド

①「ps –f」で確認。確かにPID8785が存在することがわかる



Contents

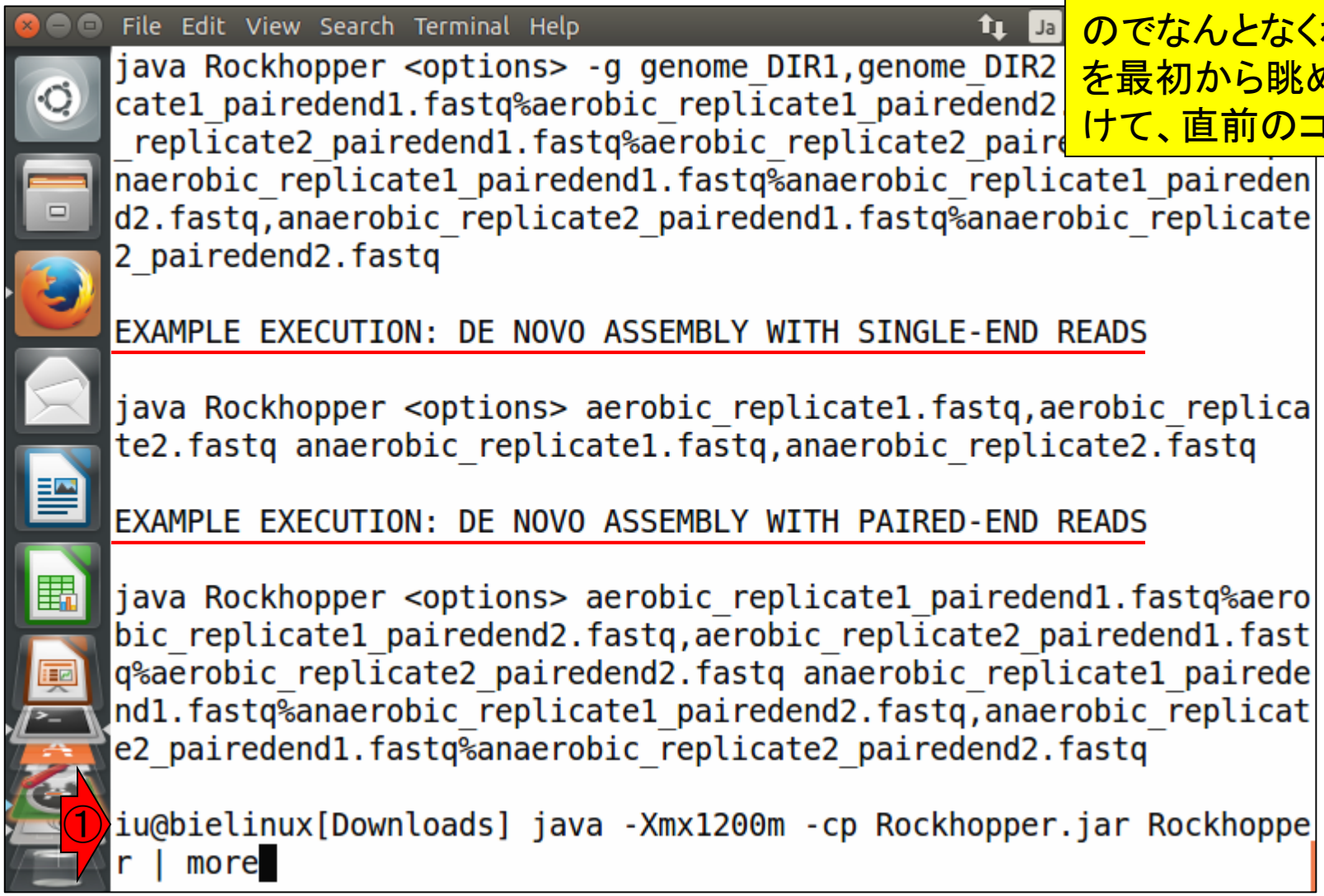
- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

W4-1：コマンドライン版

①Rockhopperのコマンドライン版を実行したい場合のやり方が書いてあるので、②その通りに実行。③リターンキーを押すと…

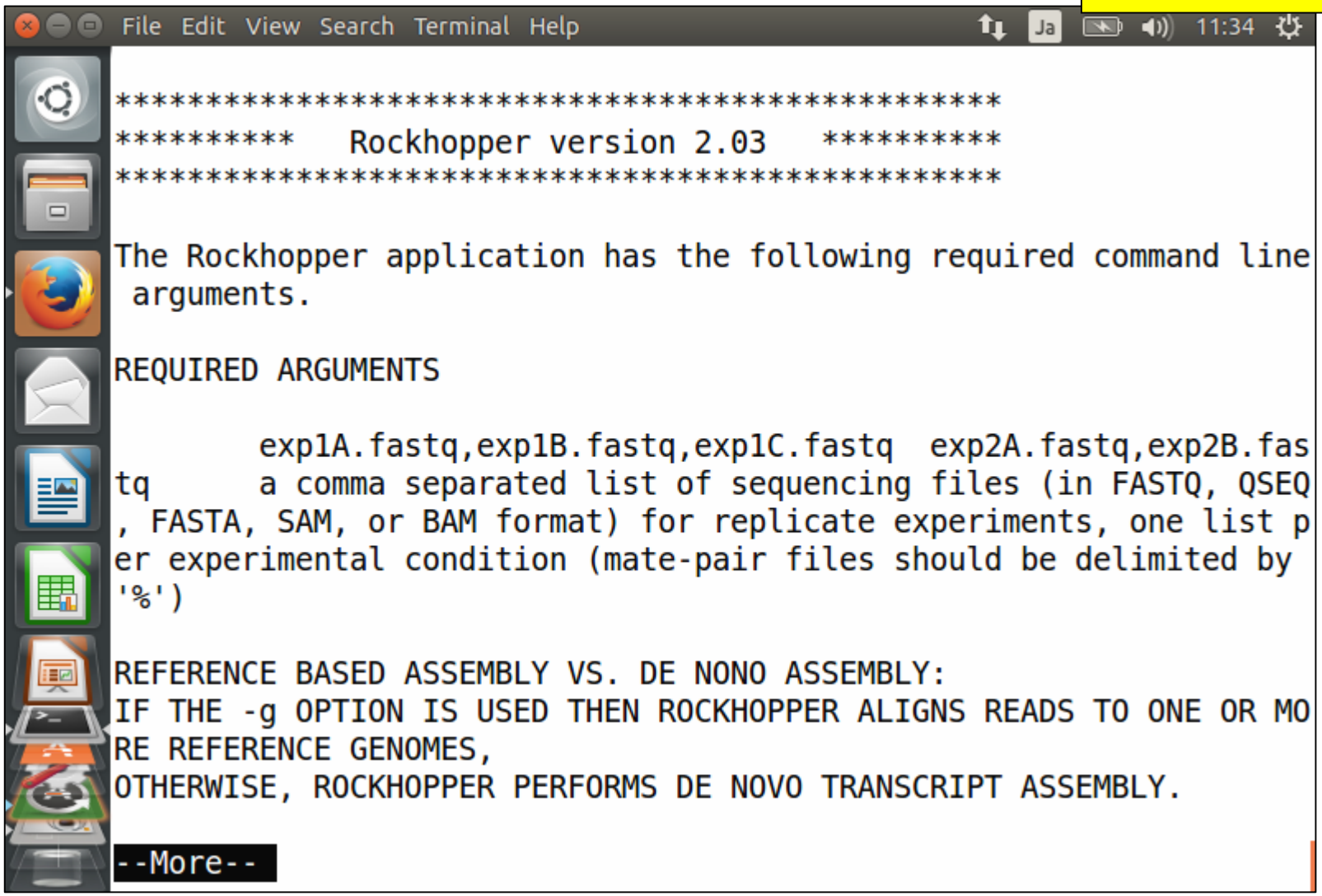
W4-1：コマンドライン版

マニュアルが一気に流れる。赤下線で示すように、最後のほうに*de novo*アセンブリのコマンド実行例があるのでなんとなくわかる。①マニュアルを最初から眺めるべく、「| more」をつけて、直前のコマンドを再実行



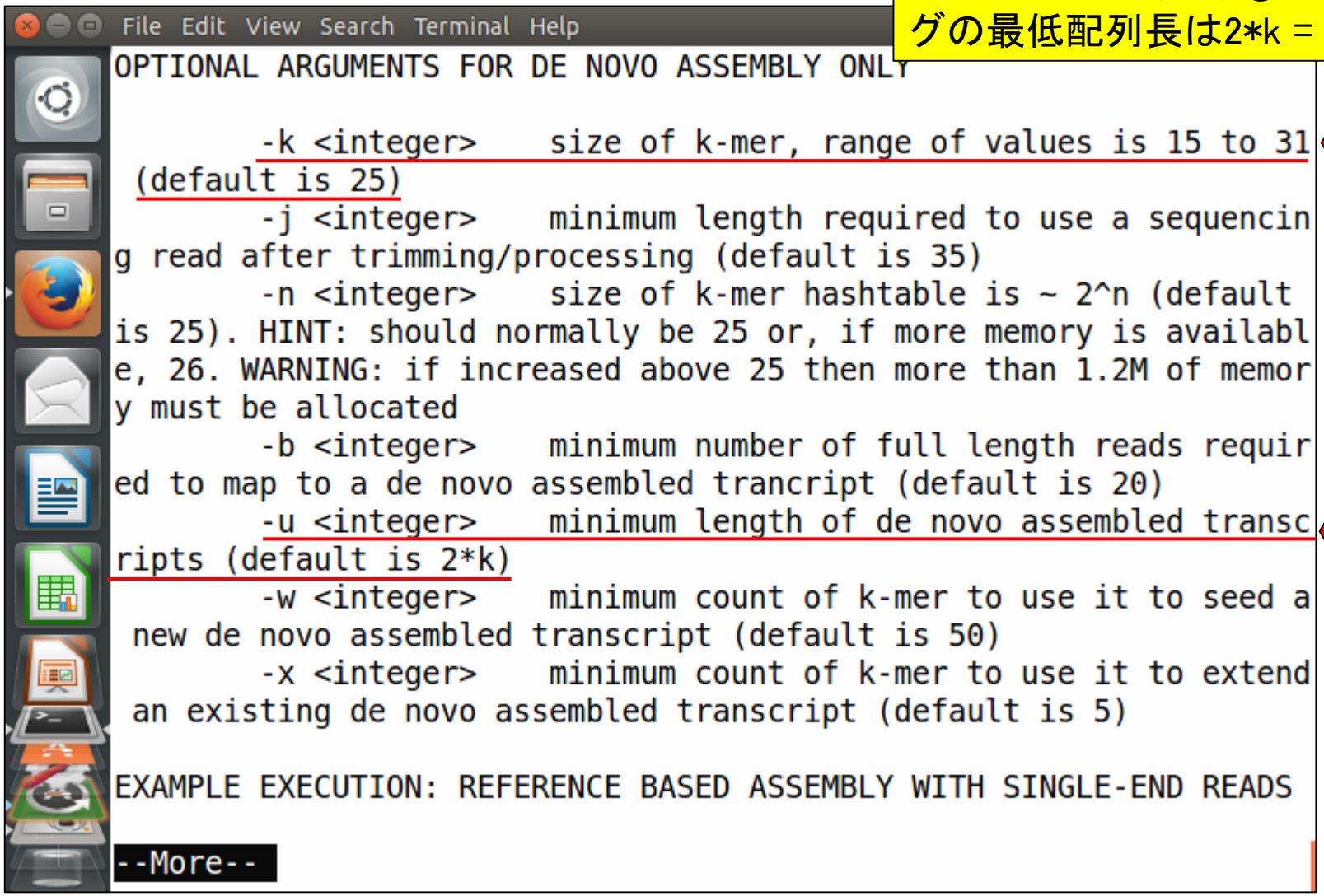
W4-1：コマンドライン版

moreコマンドは、「Returnキー」で1行分づつ、「Spaceキー」で1画面分づつスクロールできる。ただの復習



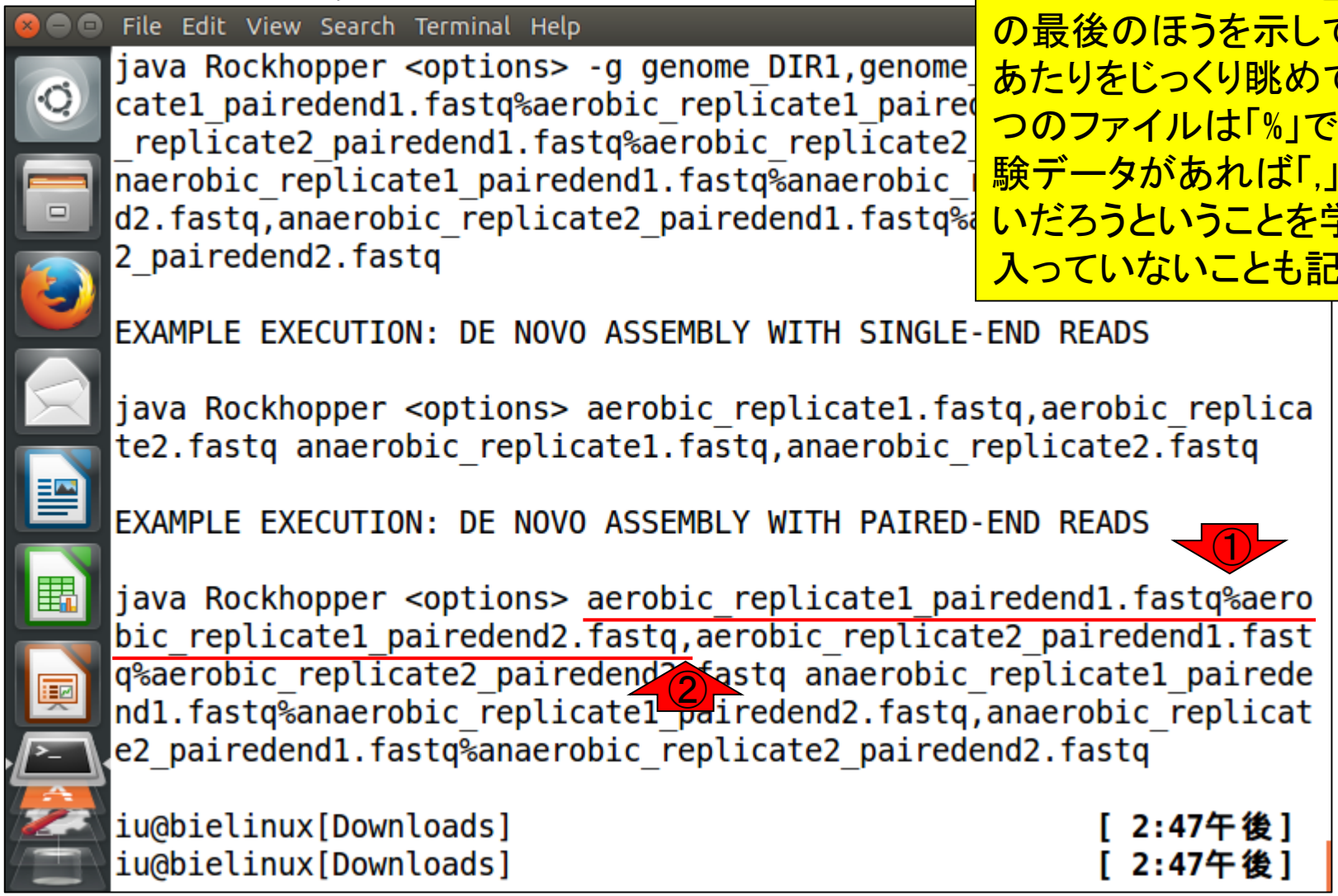
W4-1：オプション

この画面あたりがde novoアセンブリで使うオプションの説明。①kの値は、Trinityと同じく25がデフォルトのようだ。②アセンブリ後のコンティグの最低配列長は2*k = 2*25 = 50だと解釈



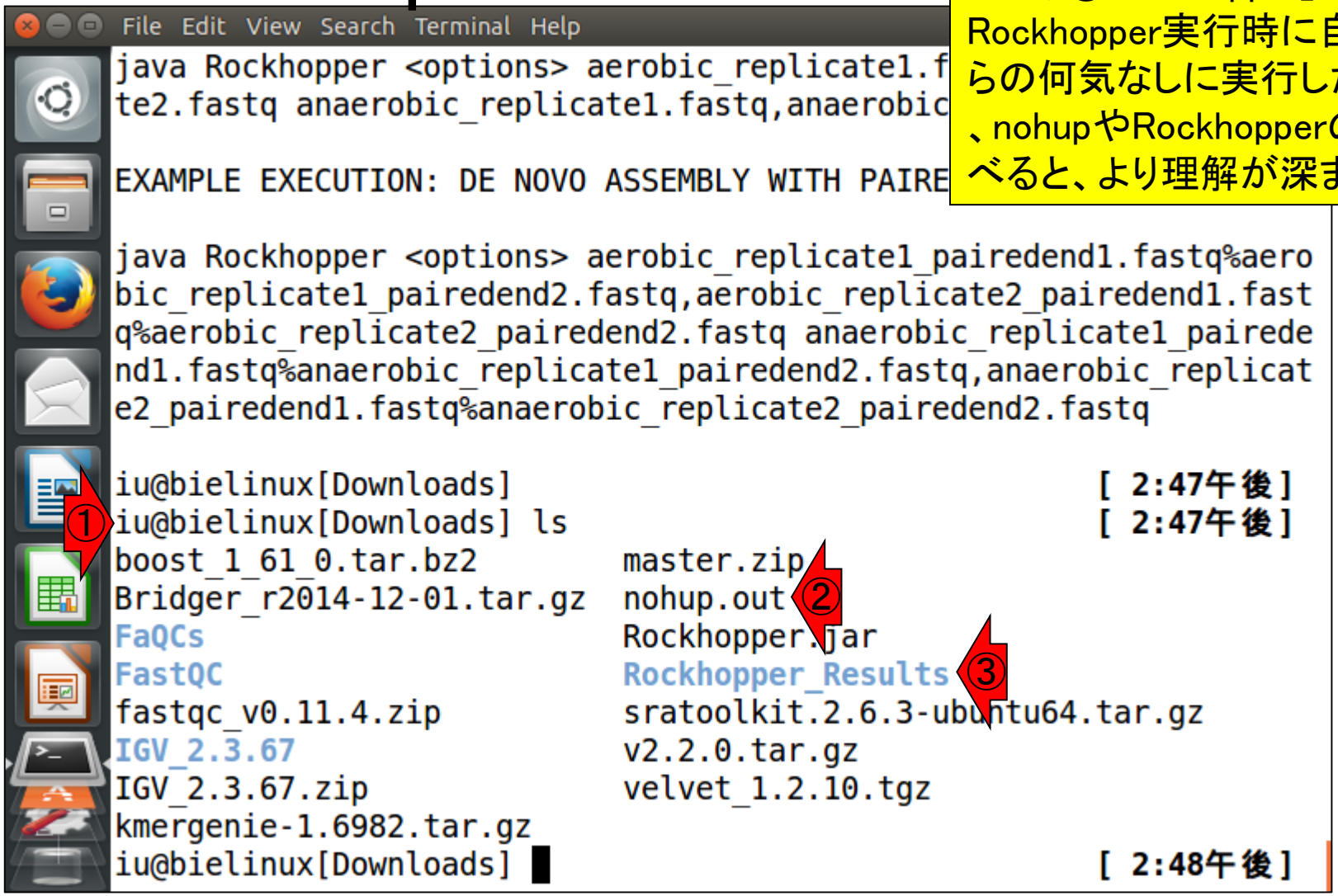
W4-2：実行コマンド例

オプションは特に指定する必要はなさそうだと解釈し、とりあえず実行例を再確認。これはリターンキーを連打してマニュアルの最後のほうを示している。赤下線部分あたりをじっくり眺めて、①paired-endの2つのファイルは「%」で連結する、②反復実験データがあれば「,」でつなげていけばよいだろうということを学習する。スペースが入っていないことも記憶にとどめておく



W4-3：Tips

①何気なしにls。②nohup.outファイルは、nohupコマンドを実行すると自動的に生成される。③Rockhopper_Resultsディレクトリは、Rockhopper実行時に自動生成される。これらの何気なしに実行したlsコマンドの結果と、nohupやRockhopperのマニュアルを見比べると、より理解が深まっていくものです



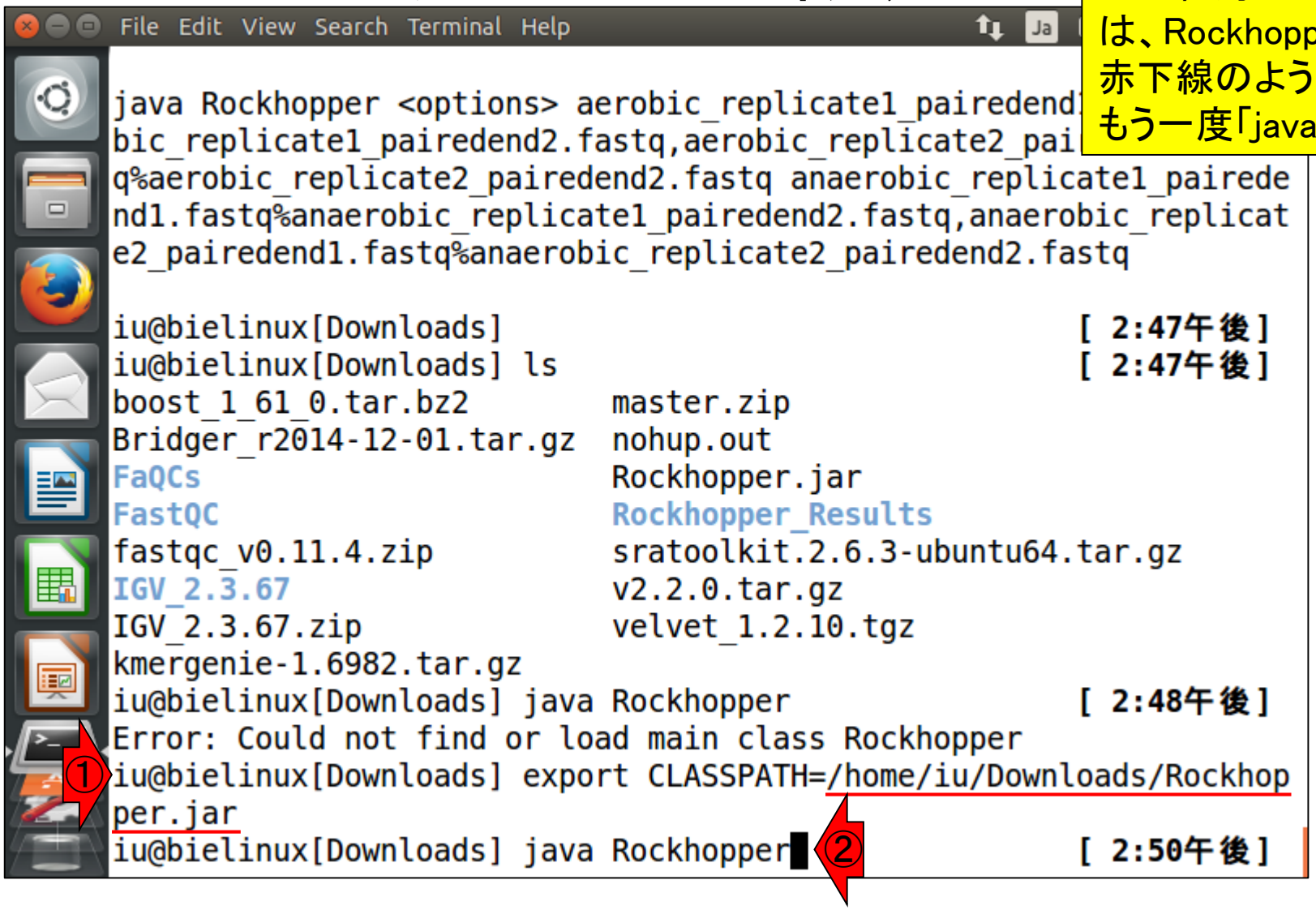
W4-4：クラスパスの設定

①EXAMPLE EXECUTIONのところを眺めると、実行は赤下線のようなコマンドを打つと書いてある。しかし、②それを実際に打ってみてもエラーが出る。理由はクラスパスの設定ができていないから

```
File  Edit  View  Search  Terminal  Help

①EXAMPLE EXECUTION: DE NOVO ASSEMBLY WITH PAIRED-END READS

java Rockhopper <options> aerobic_replicate1_pairedend1.fastq%aero
bic_replicate1_pairedend2.fastq,aerobic_replicate2_pairedend1.fast
q%aerobic_replicate2_pairedend2.fastq anaerobic_replicate1_pairede
nd1.fastq%anaerobic_replicate1_pairedend2.fastq,anaerobic_replicat
e2_pairedend1.fastq%anaerobic_replicate2_pairedend2.fastq

iu@bielinux[Downloads]                                    [ 2:47午後]
iu@bielinux[Downloads] ls                                 [ 2:47午後]
boost_1_61_0.tar.bz2          master.zip
Bridger_r2014-12-01.tar.gz    nohup.out
FaQCs                         Rockhopper.jar
FastQC                        Rockhopper_Results
fastqc_v0.11.4.zip            sratoolkit.2.6.3-ubuntu64.tar.gz
IGV_2.3.67                    v2.2.0.tar.gz
IGV_2.3.67.zip                velvet_1.2.10.tgz
kmergenie-1.6982.tar.gz
②iu@bielinux[Downloads] java Rockhopper                   [ 2:48午後]
Error: Could not find or load main class Rockhopper
iu@bielinux[Downloads]                                    [ 2:48午後]
```

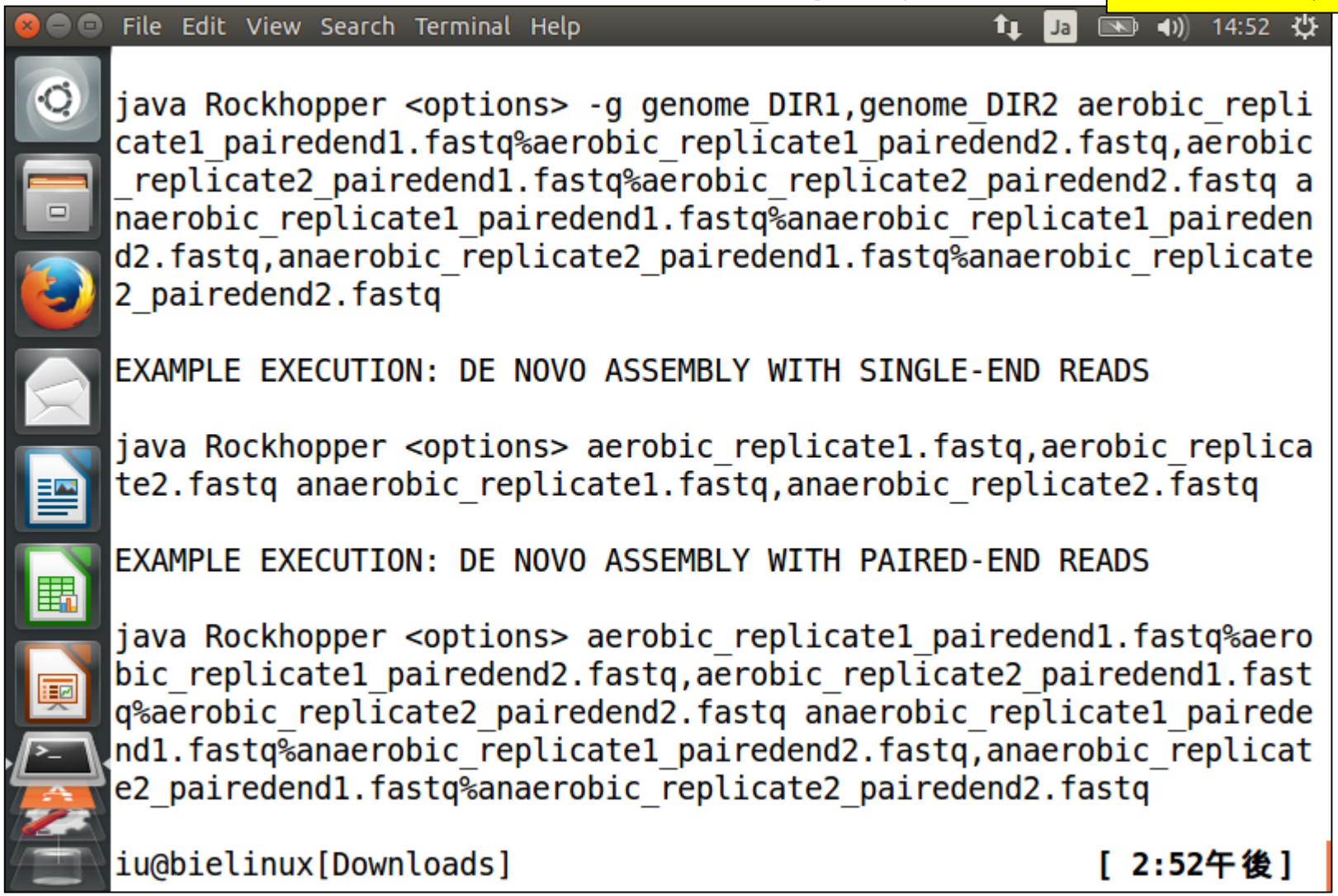
W4-4：クラスパスの設定

①クラスパスの設定は「export CLASSPATH=*設定したいjarファイルの絶対パス*」。講習会の環境では、Rockhopper.jarの絶対パスは赤下線のように書く。②設定後にもう一度「java Rockhopper」と打つ



W4-4：クラスパスの設定

リターンキーを押したあとの状態。
エラーメッセージではなく、正しくマニュアルが表示される

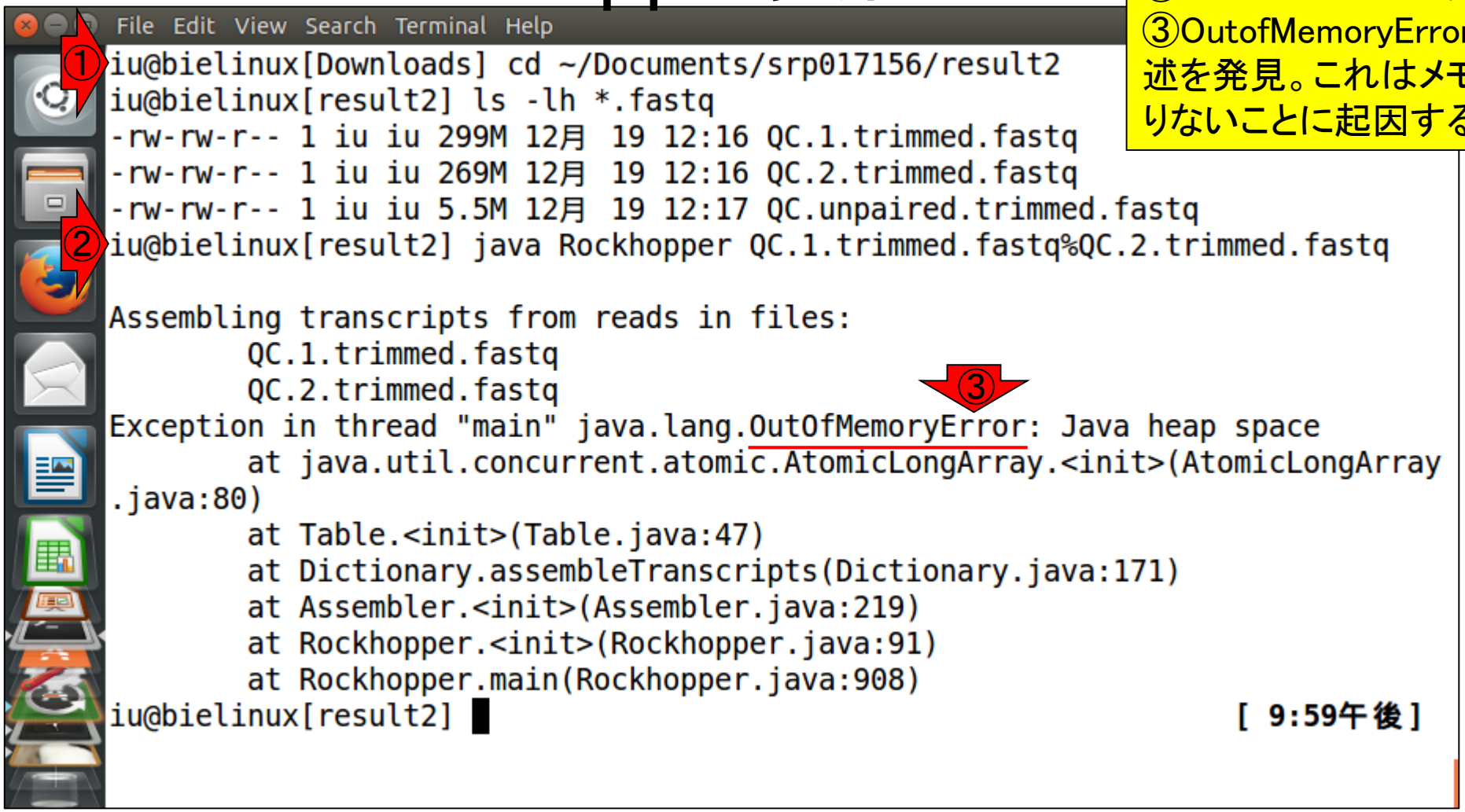


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

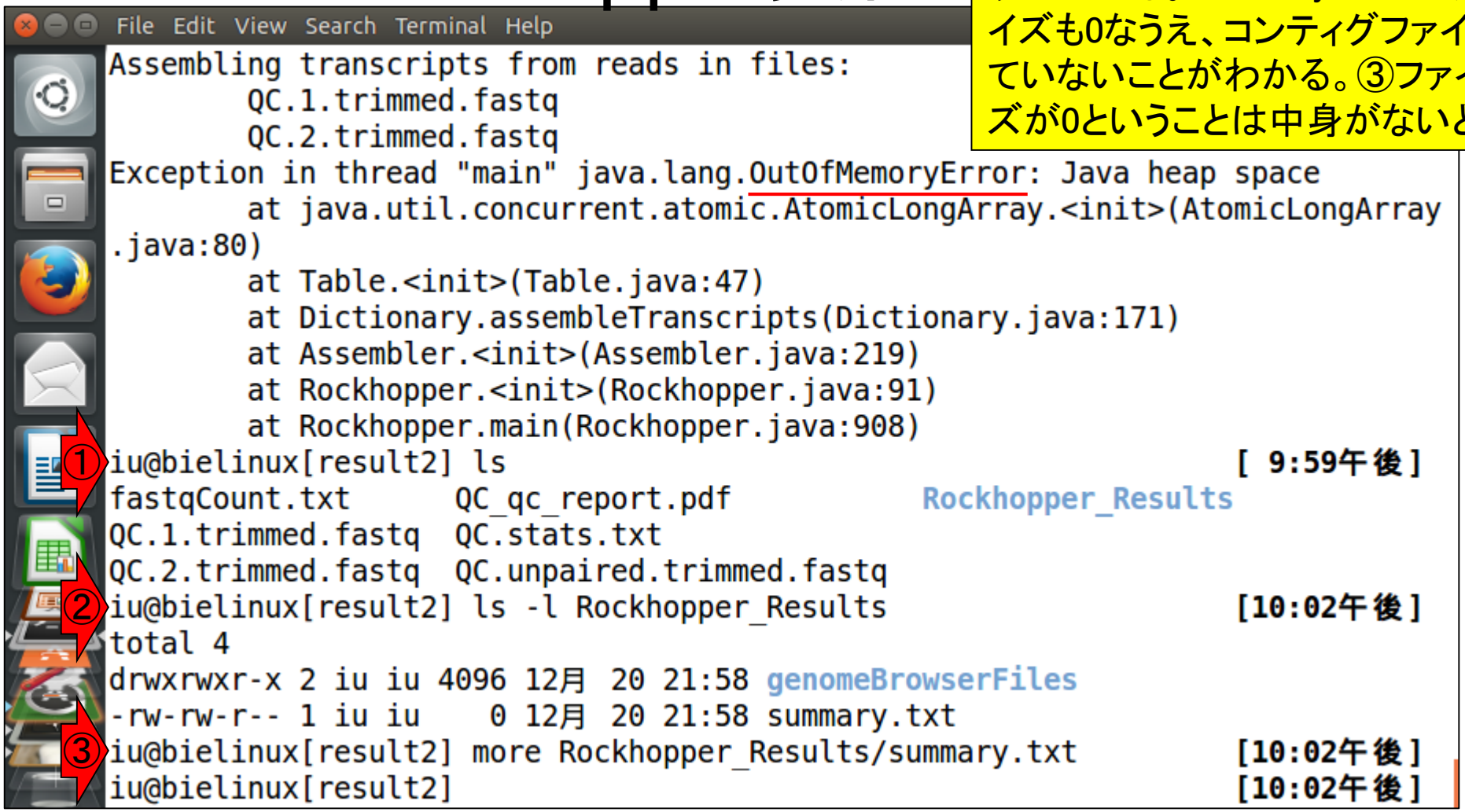
W5-1：Rockhopper実行

①FaQCs実行結果ファイルを含むディレクトリに移動して、
②*de novo*アセンブリを実行。
③OutofMemoryErrorという記述を発見。これはメモリが足りないことに起因するエラー



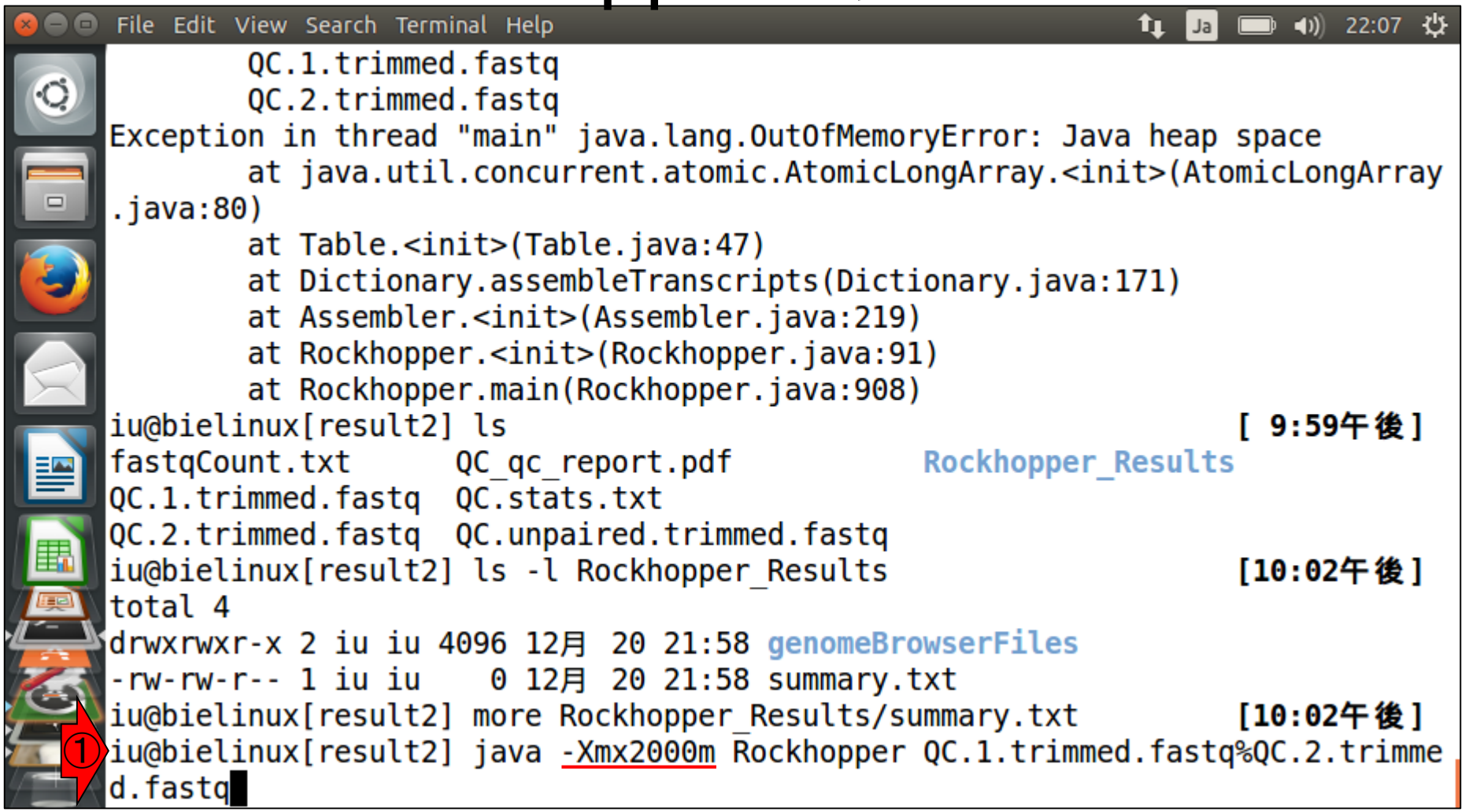
W5-1：Rockhopper実行

①lsすると、一応Rockhopper_Resultsディレクトリはできている。②その中身を眺めている。summary.txtのファイルサイズも0なうえ、コンティグファイルもできていないことがわかる。③ファイルサイズが0ということは中身がないということ



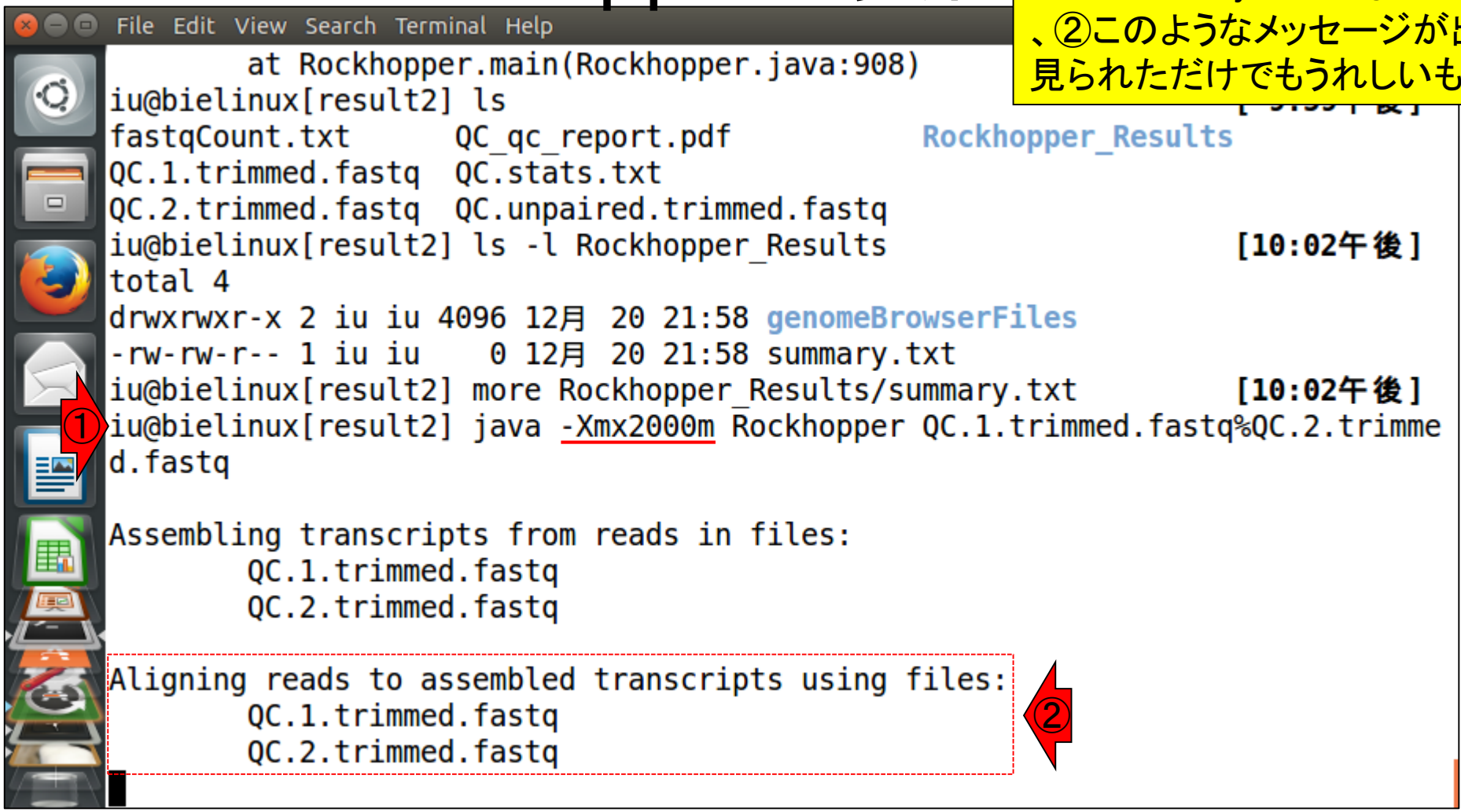
W5-2：Rockhopper再実行

①最大メモリを2GBまで増やしてリトライ。数分程度



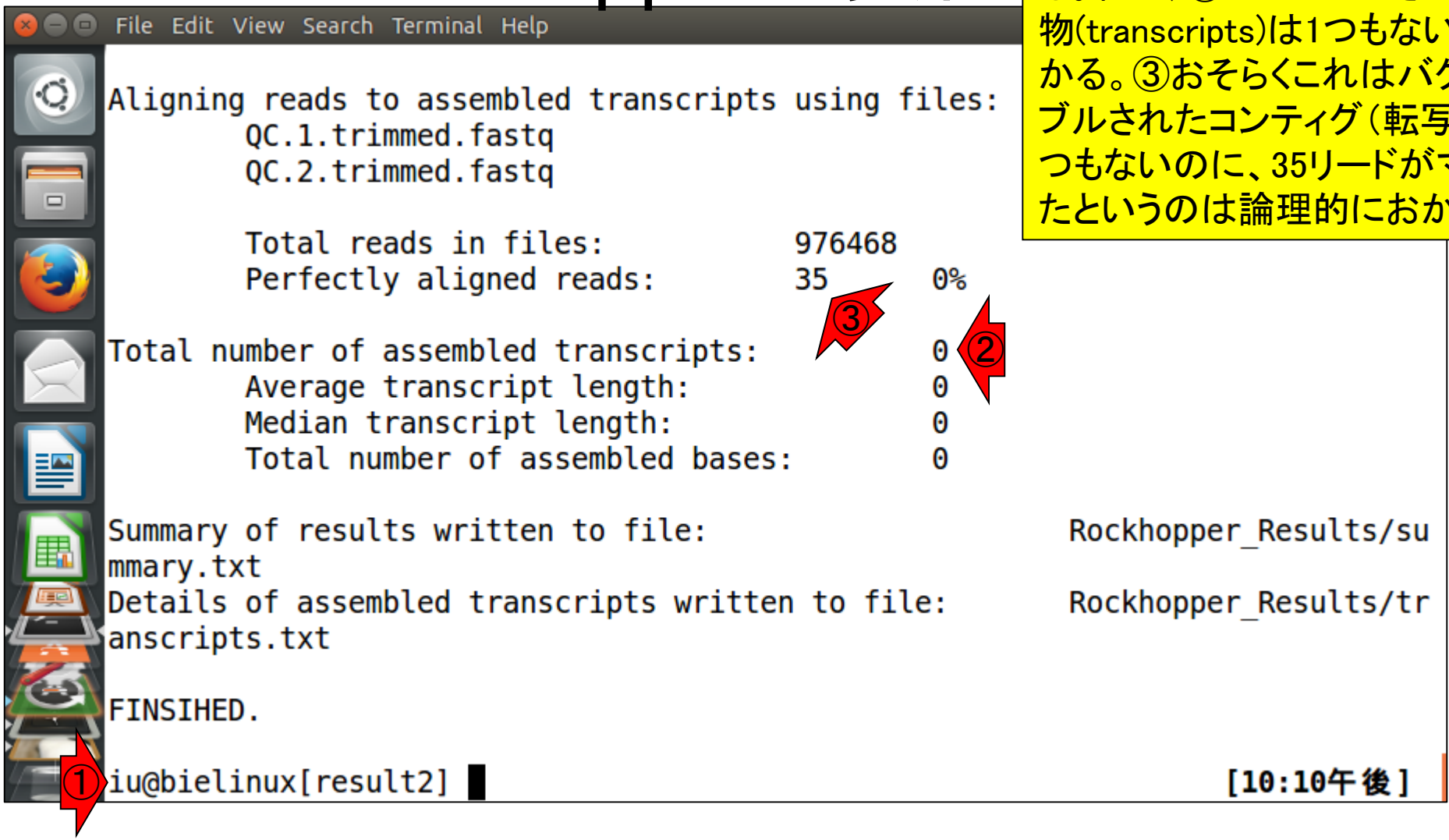
W5-2：Rockhopper再実行

①「-Xmx2000m」オプションをつける前は、赤枠の途中経過が出る前にOutOfMemoryErrorとなっていたので、②このようなメッセージが出るのを見られただけでもうれしいものです



W5-2：Rockhopper再実行

①無事*de novo*アセンブリが終了し、コマンド入力待ち状態になっている。但し、②アセンブルされた転写物(transcripts)は1つもないことがわかる。③おそらくこれはバグ。アセンブルされたコンティグ（転写物）が1つもないのに、35リードがマップされたというのは論理的におかしい



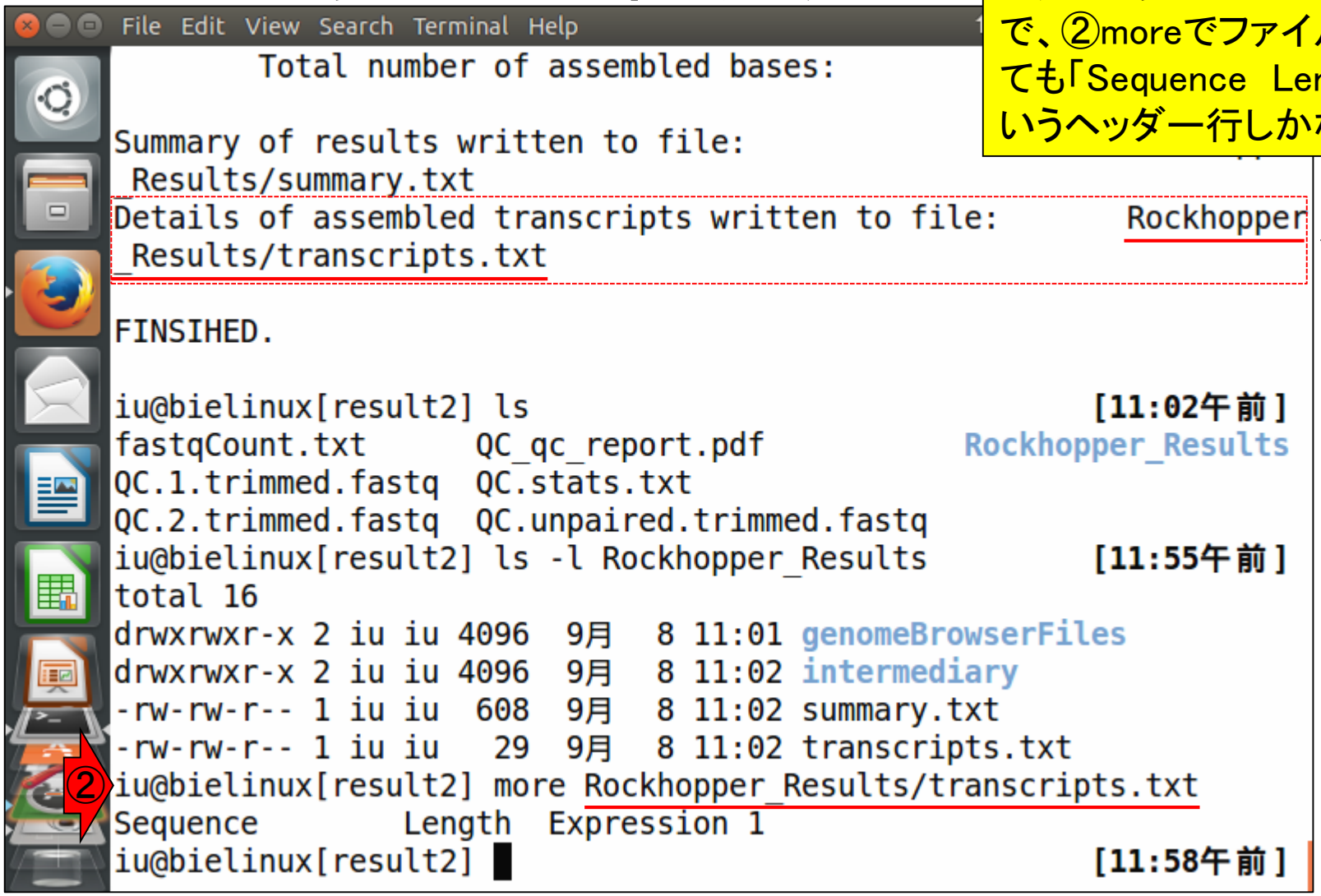
W5-3：実行結果概観

① Rockhopper_Resultsディレクトリの中身は、エラーを吐いたとき(W5-1)とは異なることがわかる。②summary.txtの中身は、赤枠でも示されているように、基本的に画面に表示されていたアセンブル結果の要約情報が含まれている

```
iu@bielinux[~/Documents/srp017156/result2]
        Median transcript length:
        Total number of assembled bases:

Summary of results written to file:                    Rockhopper_Results/su
mmary.txt
Details of assembled transcripts written to file:      Rockhopper_Results/tr
anscripts.txt

FINSIHED.

iu@bielinux[result2] ls                                              [10:10午後]
fastqCount.txt      QC_qc_report.pdf          Rockhopper_Results
QC.1.trimmed.fastq  QC.stats.txt
QC.2.trimmed.fastq  QC.unpaired.trimmed.fastq
①iu@bielinux[result2] ls -l Rockhopper_Results                       [10:17午後]
total 16
drwxrwxr-x 2 iu iu 4096 12月  20 21:58 genomeBrowserFiles
drwxrwxr-x 2 iu iu 4096 12月  20 22:10 intermediary
-rw-rw-r-- 1 iu iu  608 12月  20 22:10 summary.txt ②
-rw-rw-r-- 1 iu iu   29 12月  20 22:10 transcripts.txt
iu@bielinux[result2]                                                 [10:17午後]
```

W5-3：実行結果概観

①アセンブルされた転写物配列情報はtranscripts.txtファイルに格納される。ただし、この場合は1つもコンティグがないので、②moreでファイルの中身を表示させても「Sequence Length Expression 1」というヘッダー行しかないことがわかる

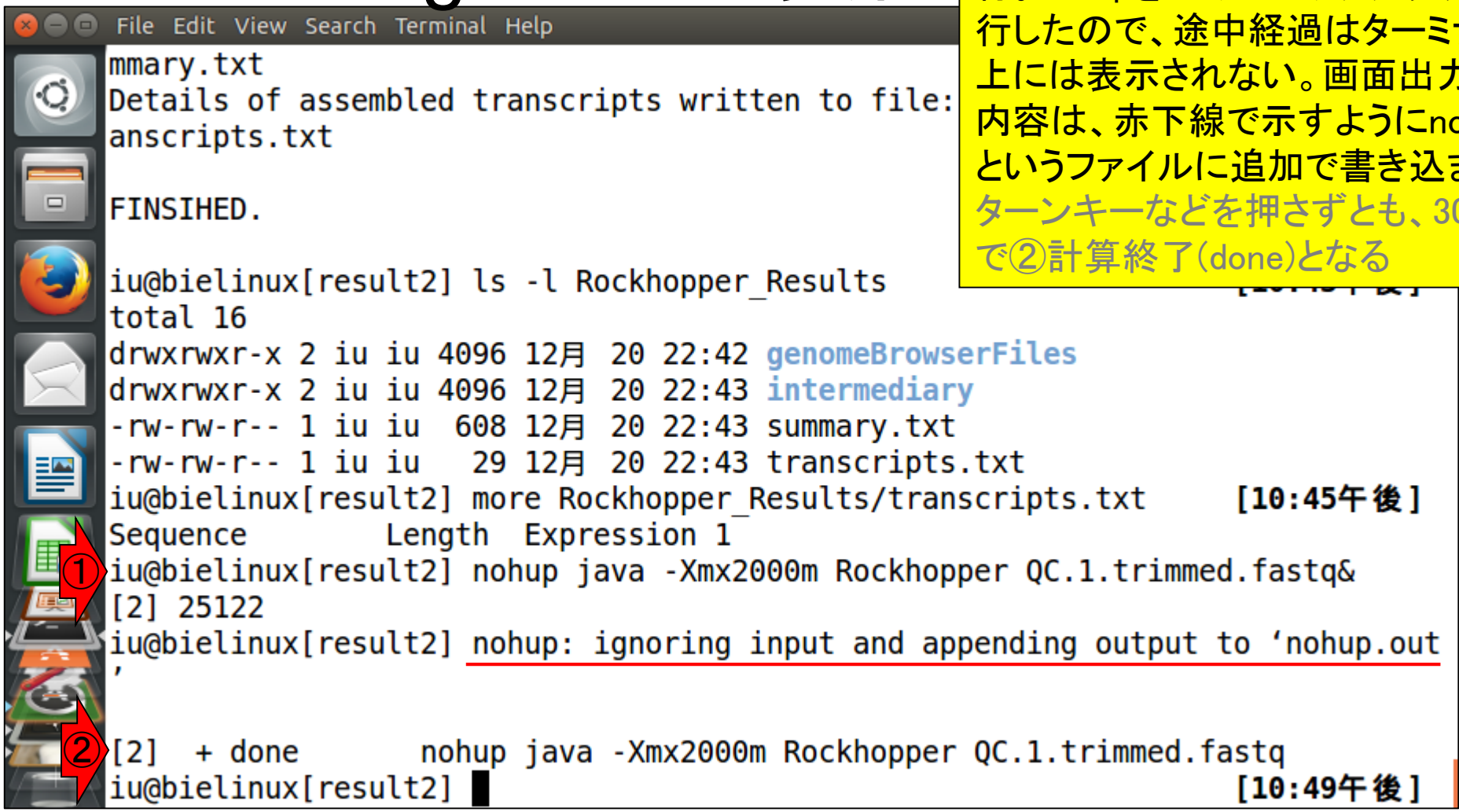


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

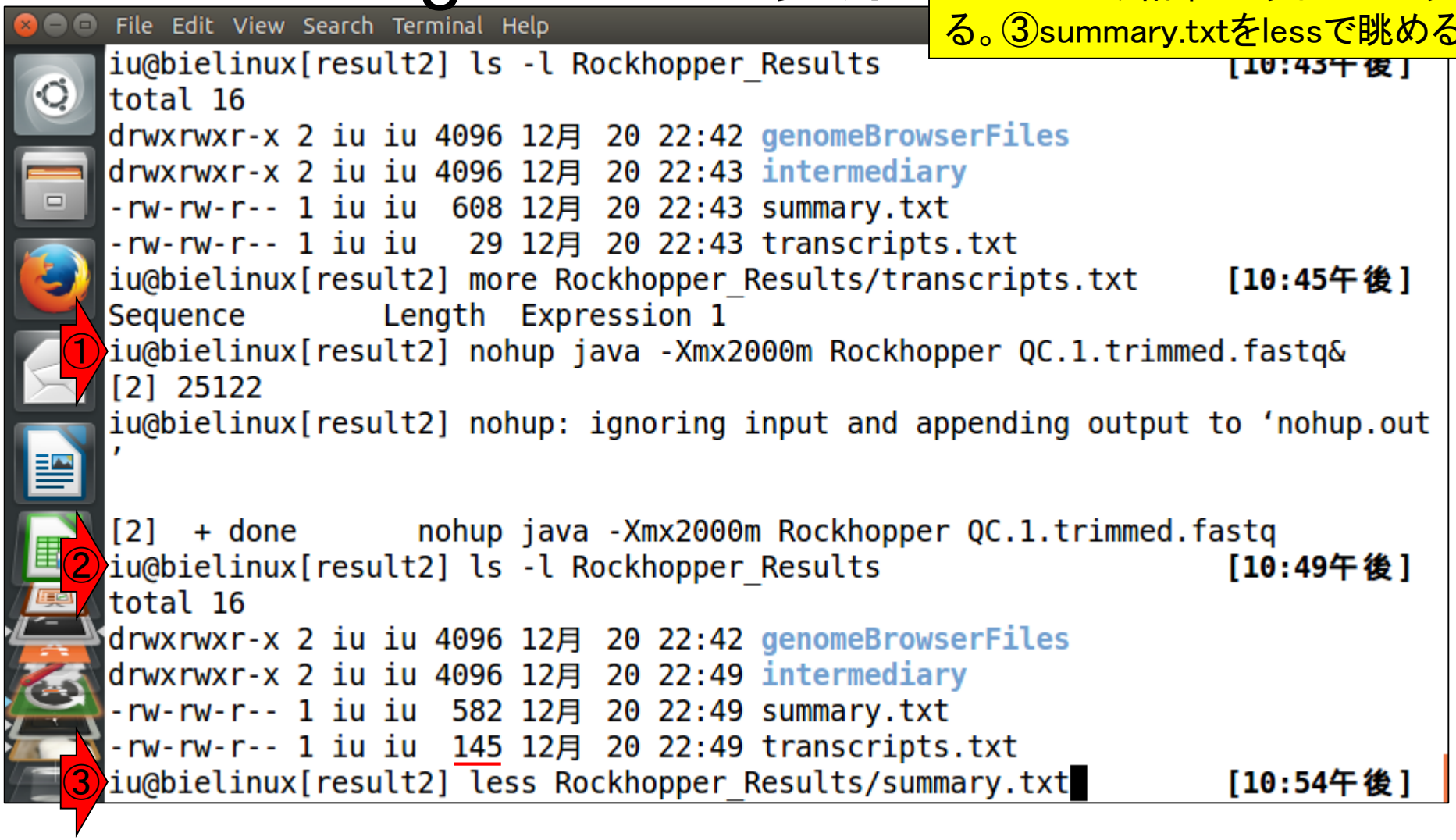
W6-1：single-endで実行

①single-endとしてforward側のみのファイル(QC.1.trimmed.fastq)を入力として実行。nohupをつけてバックグラウンドで実行したので、途中経過はターミナル画面上には表示されない。画面出力される内容は、赤下線で示すようにnohup.outというファイルに追加で書き込まれる。リターンキーなどを押さずとも、30秒ほどで②計算終了(done)となる



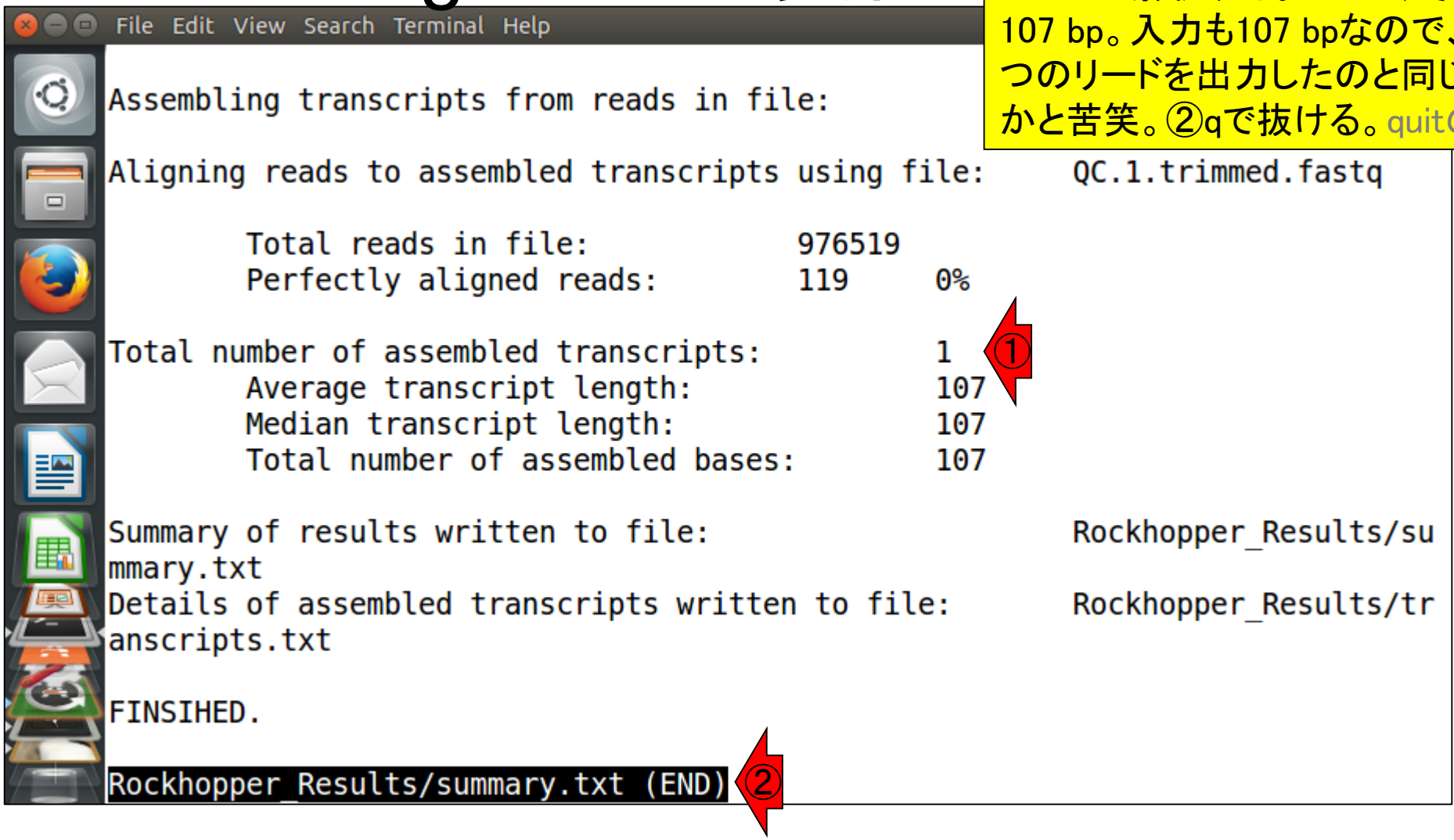
W6-1：single-endで実行

②Rockhopper_Results中のtranscripts.txtが145 bytesとなっていることから、何かしらアセンブリ結果があるのだろうと解釈する。③summary.txtをlessで眺める



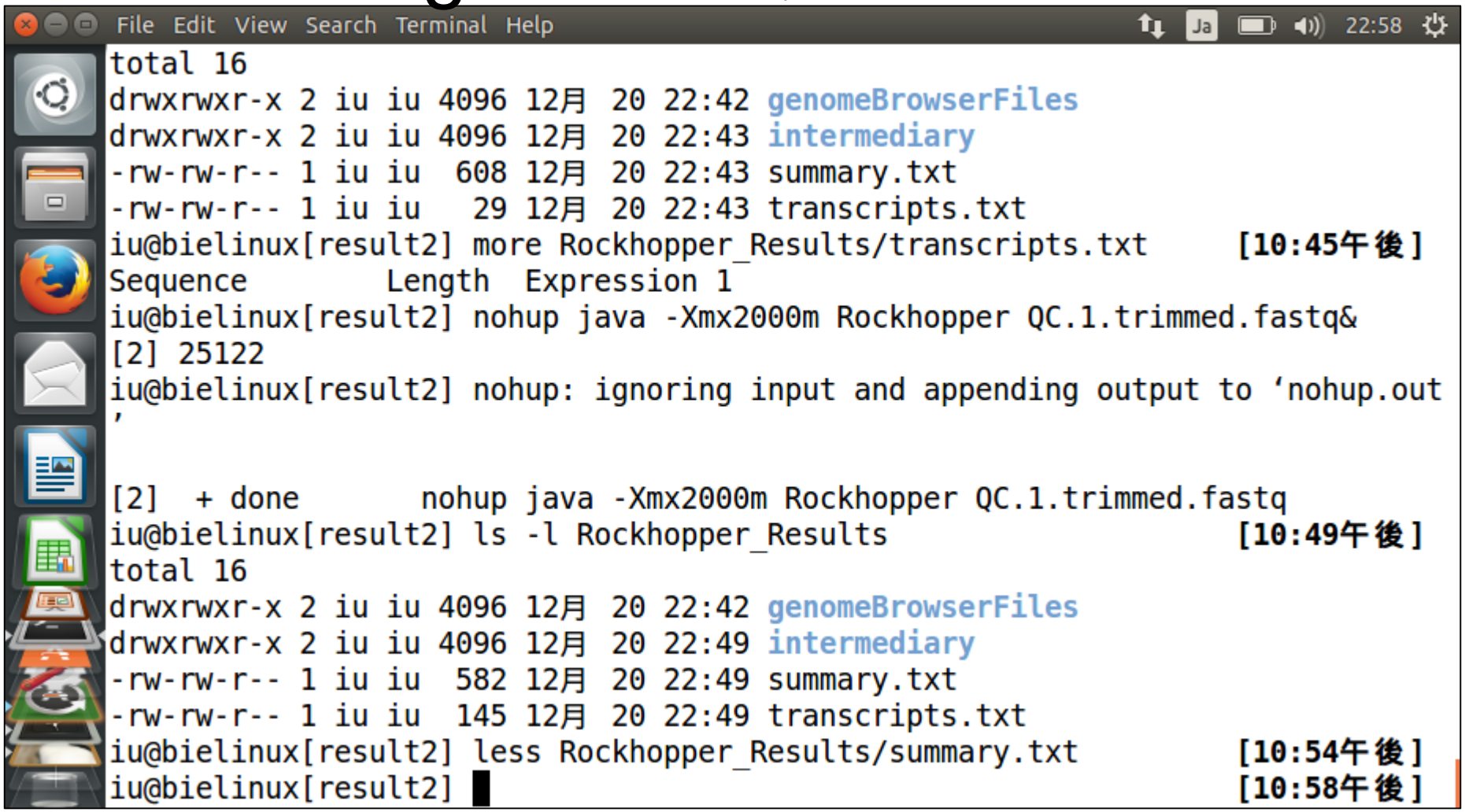
W6-2：single-endで実行

summary.txtのless実行結果。①アセンブル結果として転写物が1つだけ得られたと解釈する。しかし、その長さは107 bp。入力も107 bpなので、どれか1つのリードを出力したのと同じじゃないかと苦笑。②qで抜ける。quitのqです



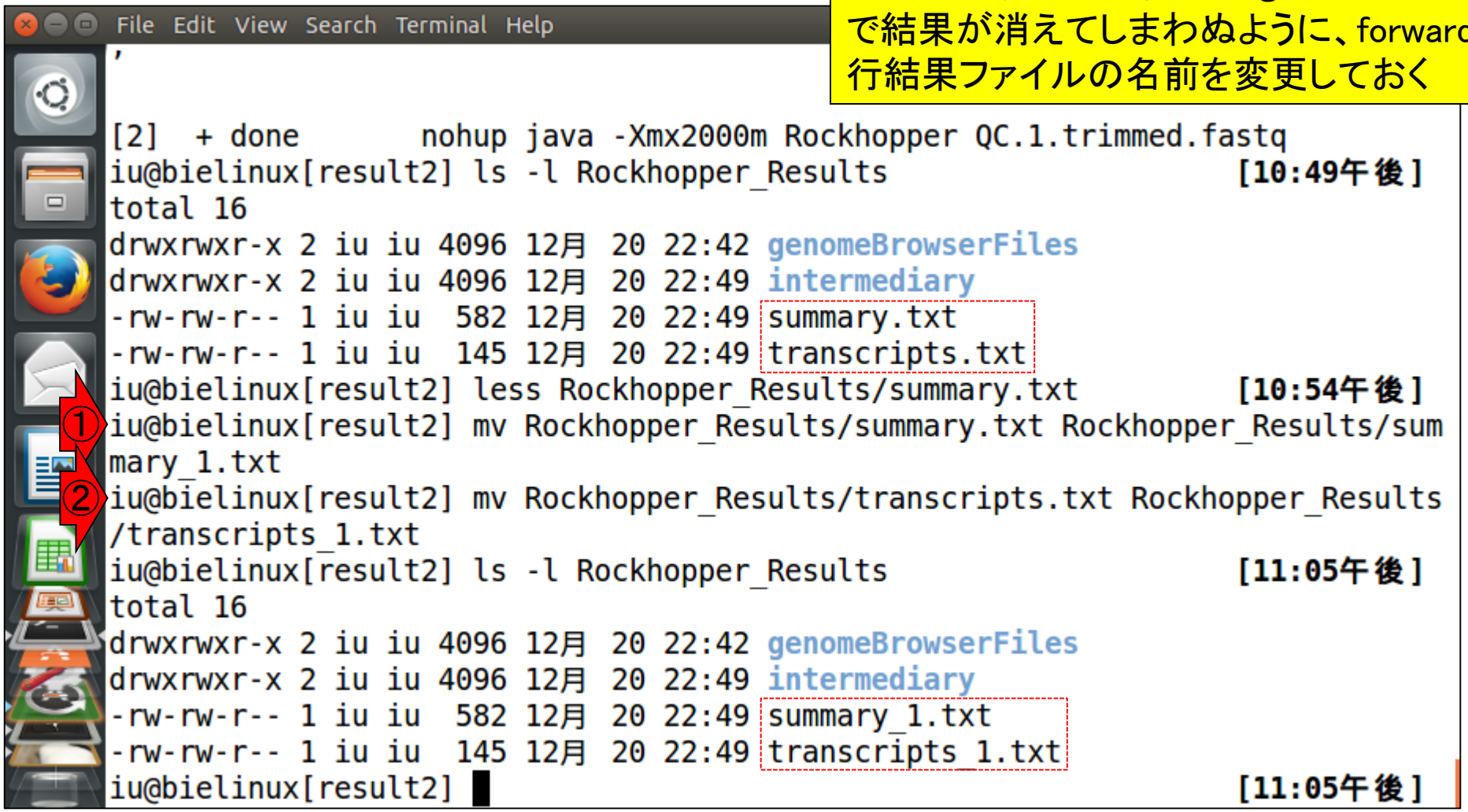
コマンド入力待ち状態に戻る

W6-2：single-endで実行



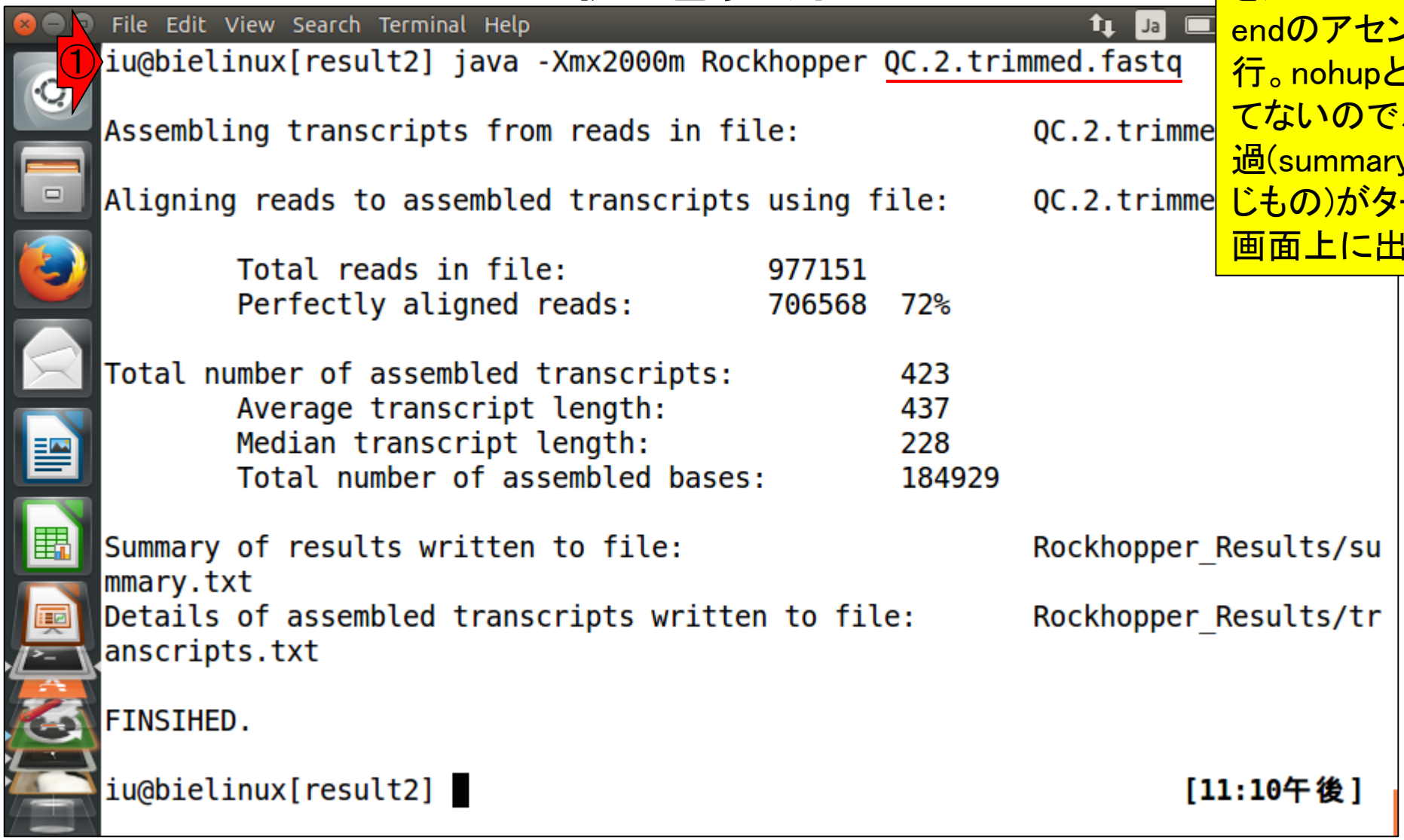
W6-3：mvでrename

ここまでの作業で、Rockhopperはアセンブリ結果ファイルを上書き保存していることがわかる。この後に行うreverse側のsingle-endのアセンブリで結果が消えてしまわぬように、forward側の実行結果ファイルの名前を変更しておく



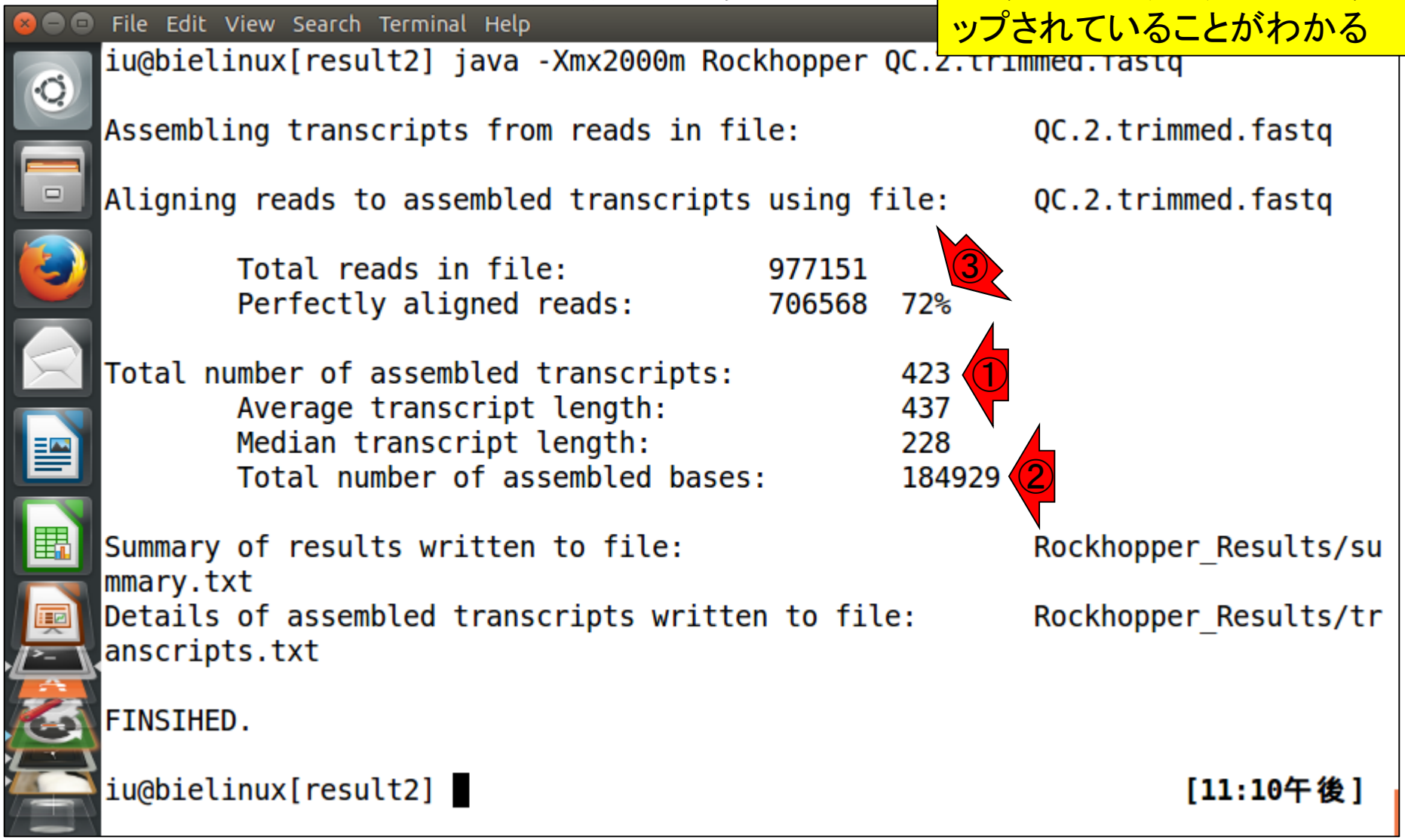
W6-4：reverse側を実行

①reverse側ファイル(QC.2.trimmed.fastq)を入力としてsingle-endのアセンブリを実行。nohupと&をつけてないので、途中経過(summary.txtと同じもの)がターミナル画面上に出力される



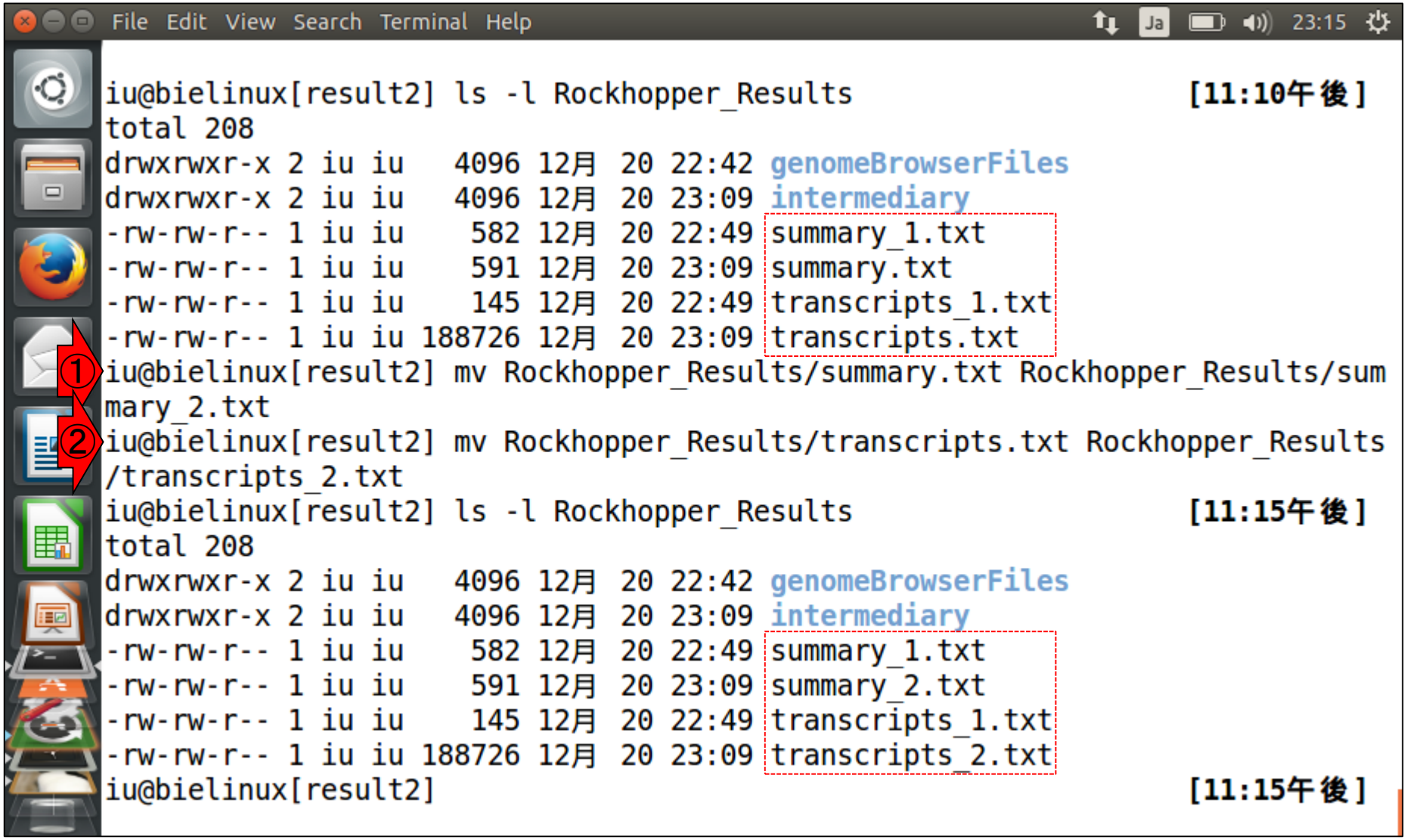
W6-4：reverse側を実行

①アセンブルされた転写物数は423個!
②総塩基数は184,929。③入力リード数977,151個のうち、72%（706,568個）がマップされていることがわかる



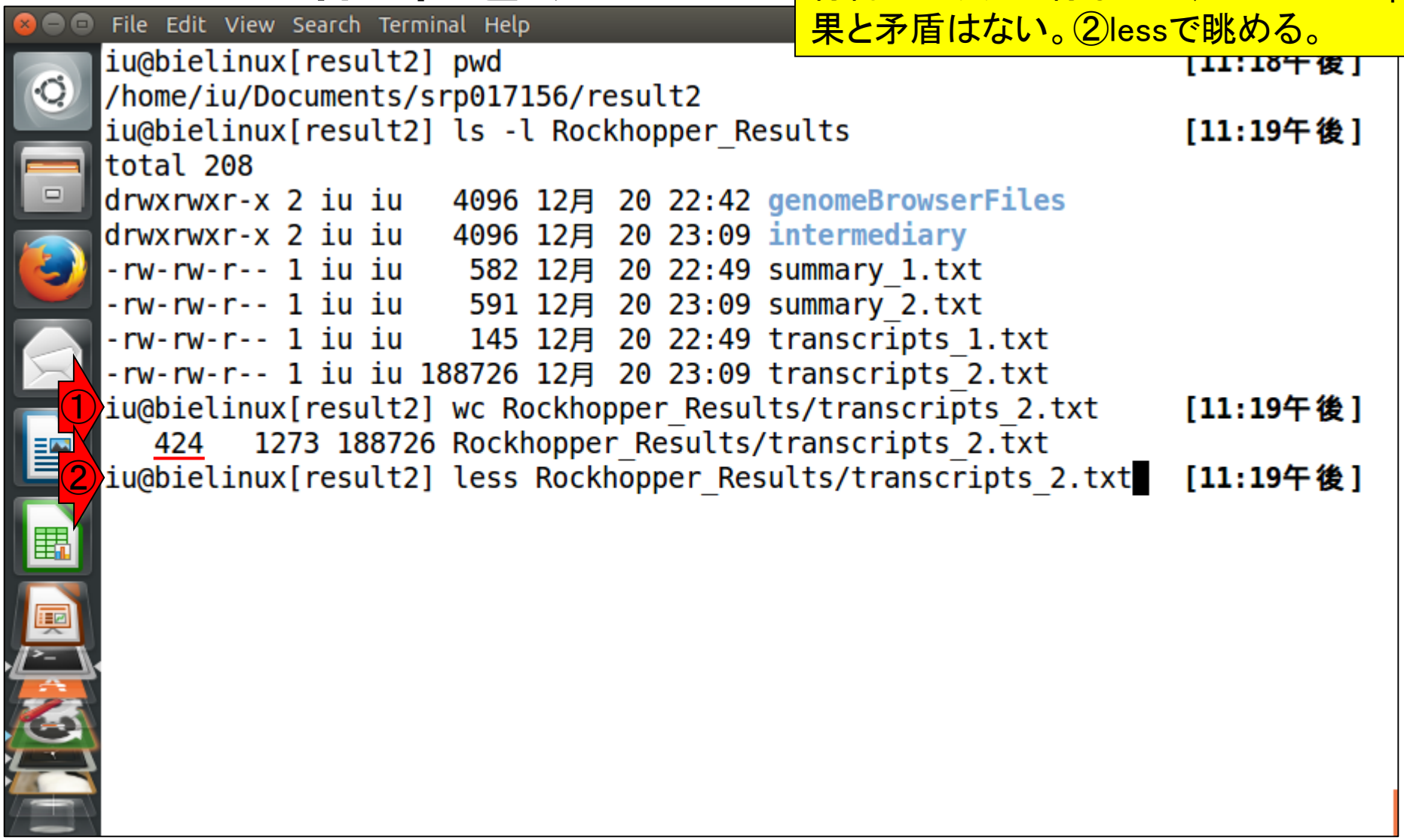
W6-4：mvでrename

念のため、reverse側の実行結果ファイルの名前を*_2.txtに変更しているだけです



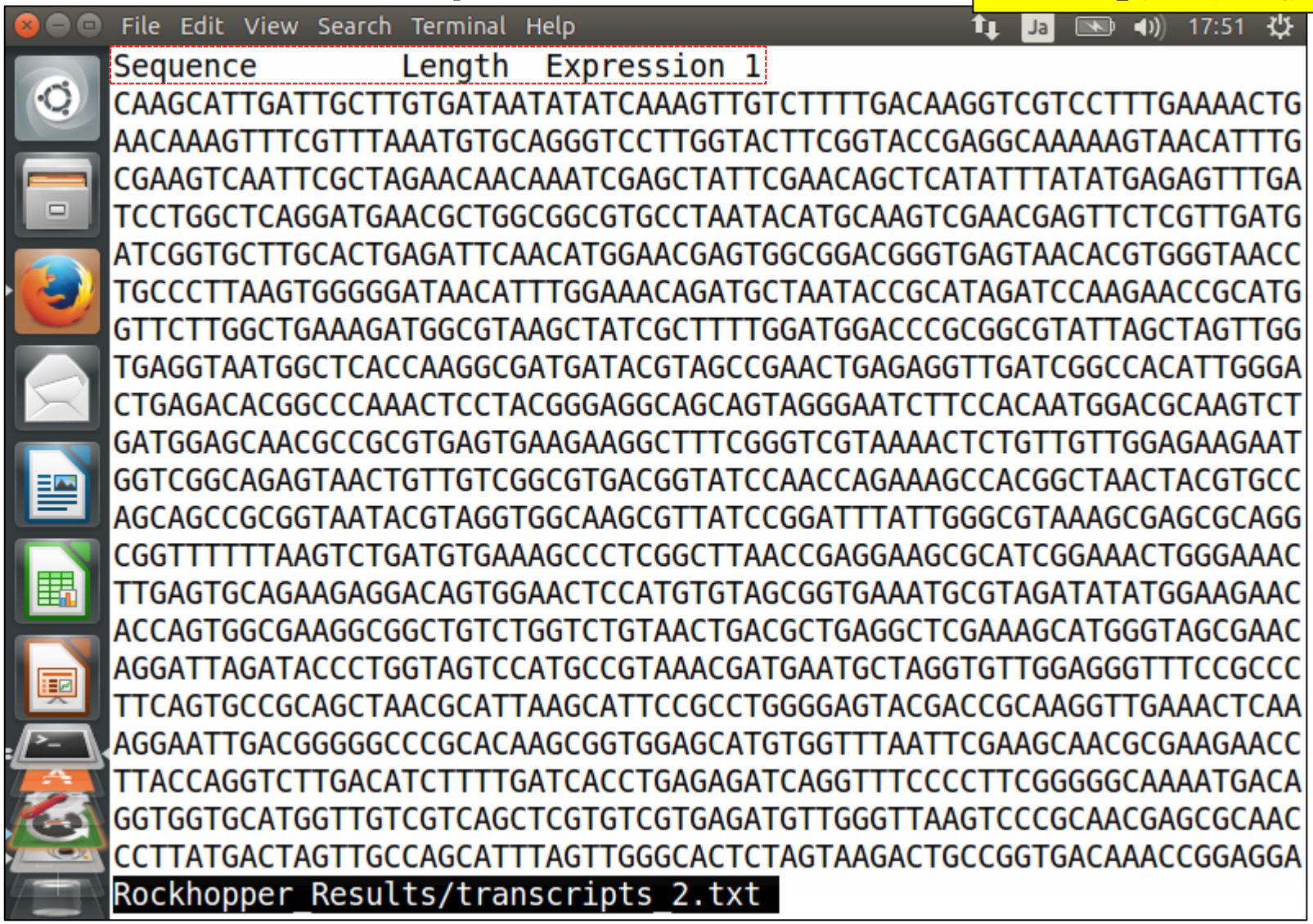
W6-5：結果を眺める

①reverse側の実行結果ファイルtranscripts_2.txtの行数をwcで調べる。424行だったが、最初の1行目はヘッダー行なので、423 transcriptsの結果と矛盾はない。②lessで眺める。



W6-5：結果を眺める

①lessで開いた直後の状態。赤枠部分がヘッダー行。ファイル末尾に移動したい場合は「G」、先頭に移動したい場合は「g」



W6-5：結果を眺める

ファイル末尾に移動。赤枠内に見えているのは、計4 transcripts分の情報。2列目がLengthなので、赤下線部分が転写物の長さ情報に相当。qで抜ける。
①出力ファイルの説明はRockhopperのUser Guide

File Edit View Search Terminal Help

```
          128       113
TATGCTGATCGGGAAATTTCTGAATTATCCGGTGGTCAACAACAGCGAGTTGCCATTGCTCGAGCG
ATTGTACTCGAGCCGCAAGTATTGCTGCTAGATGAACCGTTATCAGCACTTGACGCCAAATTGCGT
AAGGATATGCAATATGAATTGCGCGAATTGCAGGAACGGTTGGGGATCACTTTCCTATTTGTGACA
CACGACCAAGAAGAAGCGTTGGCTTTGTCGGACGAAATTTTTGTCATGAACGATGGTGAAGTGCAA
CAAAGTGGTACGCCAGTTGATATTTATGATGAGCCGGTCAATCATTTTGTGGCGGATTTCATTGGT
GAAAGTAACATCATTCAAGGGCACATGATTAAGGACTTTTTAGTTGAGTTCAATGGCAAACGGTTT
GAATGTGCCGATGCCGGAATGCG       419       138
TCACTCAACCAACCCCGCCAAAGACGGCGAAACGGTTGGCTTGACGTTTGATCCTGAGGACATCCA
TGTCATGCGGCTTAACGAATCTGAAGAAGATTTCGACGCTCGGCTGGAAACCTACGAAGGGGAATA
ACGTCAGTAAGGATCAGTTGGGGCAGTGTCATCGAAGCGTTTTGACGATTAGATAGACACTGGTCG
GCGCTGATCGATTTTCGAACTGAGGAGGGTCATCGTGAAAAAATCCACCACAAACGCCGCATTCTA
CACACCTTATGTGATGTGGCTGGCATTATTTGTGA       299       162
CTTGCCAATTATCAGACCTATTTTCAATCAGGCACTTATATTATGATGACGATCAATTCCGTCTGG
TATGCGTTCCTGATCACGTTAGCAACTTTGTTAATCAGCTATCCAACAGCATATTTGCTGCATTAT
GCCAAGCATAAACAGTTATGGCTGTTGCTGATTATTCTACCGACTTGGATTAAC       186
      139
AATAGTTTCTTGGGAATGTTTGGCATCGCGCCCCAGCAATTCCTGTTCACGGATTTTAGTTTCATT
TTTGTTGCTGCTTATATTGAAATCCCGTTCATGATCTTGCCAATTTTCAATGCAATCGAAGAATTA
CCGGAAAACCTTGTCAATGCTGCCCAAGATTTAGGGGCCAAAAGCTGGCAGACCTTCACCAAGGTG
ATCTGGCCGTTGACAATTTCTGGCGTGAAATCCGGGGTTCAAGCTGTTTTCATTCCAAGTTTGAGT
TTATTCATGT       274       121
(END)
```

HOME
Download
User Guide ①
FAQ

第5回原稿PDFのp195

ここまでが、バクテリア用*de novo*トランスクリプトームアセンブラである①Rockhopper2実行部分

として、FastQC によるクオリティチェックを行えばよい［W1-2］。著者らは、FastQC 実行結果ファイルの項目（Overrepresented sequences）を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している［W1-3］。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ（assembly）であり、再構築を行うプログラムをアセンブラ（assembler）という。デノボ（*de novo*）という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」という意味である。このため、リードのみを入力として（つまり他の情報を一切利用せずに）アセンブルする際には、*de novo* assembly という表現がなされる。トランスクリプトームアセンブリとは、アセンブル対象がゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブルする場合は、*de novo* transcriptome assembly などと呼ばれる。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

①

ここまでのまとめ

①の結果に苦悩しつつ、この乳酸菌株(*Lactobacillus casei* 12A)のゲノム配列が存在することは知っていた。当時は、RのQuasRパッケージでこのゲノム配列にマップしてみたら何かわかるかも…という程度の軽いノリでした。結果的に、QuasRによるマッピング結果を眺めることで問題解決に至りました。このあたりが…

- オリジナル(SRR616268)
  - 乳酸菌paired-end RNA-seqデータで、最初の100万リードのみ抽出
  - forward側(SRR616268sub_1.fastq.gz)のリード長は107 bp
  - reverse側(SRR616268sub_2.fastq.gz)のリード長は93 bp

- FaQCs実行結果(W1-1)
  - 1,000,000リード → 977,202リード (W1-3)
  - forward側(QC.1.trimmed.fastq)
  - reverse側(QC.2.trimmed.fastq)
  - リード長はバラバラ。FastQC上で見られるIllumina adapterは消滅状態
- *de novo*トランスクリプトームアセンブリ(Rockhopper 2)実行結果
  - paired-end(QC.1.trimmed.fastqとQC.2.trimmed.fastq):0 transcript or contig (W5-2)
  - single-end (forward側のみ; QC.1.trimmed.fastq):1 transcript (W6-2)
  - single-end (reverse側のみ; QC.2.trimmed.fastq):423 transcripts (W6-4)

①

第5回原稿PDFのp197

第5回原稿p197の右側で述べているところです。特に①の赤枠あたり

イルが得られ、これが実行ファイルになる。つまり、基本的にJavaファイルのダウンロード完了がインストール完了を意味する［W2-3］。これはWindows版（Rockhopper.exe）やMacintosh版（Rockhopper.dmg）についても同じである。Bio-Linux 8では、GUI版とコマンドライン版の両方が利用可能であり、基本的に指示された通りのコマンドを打てばよい［W2-4］。バックグラウンドジョブ（nohupと&の付加）やプロセス管理（psとkill）は、特に遺伝研スパコンなどの大型計算機にセキュアシェル（secure shell; SSH）経由でログインして解析する際に利用すると思われる。このため、GUI版の起動説明（java -Xmx1200m -jar Rockhopper.jar）と絡めて、これらの基本的な利用法を示した［W3］。

コマンドライン版の実行コマンド（java -Xmx1200m -cp Rockhopper.jar Rockhopper）も、GUI版と似ている［W4-1］。「-Xmx1200m」は、最大メモリを1200MB分確保するという意味である。「-cp」は、クラスパス（classpath）を意味し、「-classpath」と書いてもよい。これは、「パスを通す」ことと本質的に同じ概念である。しかしながら、第4回（W9-5；W15-5；W18-3）で示したような「sudo ln -s /home/iu/Downloads/Rockhopper.jar /usr/local/bin」を実行してもRockhopper.jarのタブ補完がうまくいくようになるだけである。この作業では、コマンドライン版をうまく実行できない。RockhopperのEXAMPLE EXECUTIONは「java Rockhopper <options> …」となっているが、「java Rockhopper」でエラーが出ないようにするには、クラスパスを正しい手順で設定する必要がある［W4-4］。Java特有の概念であること、Rockhopperうに、アセンブルされた転写物は1つもなかったことがわかる。この原因は、前述のようにforward側の107 bpのリードファイル（QC.1.trimmed.fastq）にある。特に、100–107塩基付近に乳酸菌に由来しないもの（以下、$f_{100-107}$）がトリムしきれずに多く残っているためである。ただし、これは乳酸菌ゲノム配列にQuasR[19]を用いてリードのマッピングを行った結果（後述）を眺めることで後に判明したことである。

アセンブル結果のみを眺めていた当時は、single-endのみで実行した結果よりもpaired-endの結果のほうが悪いという、理解に苦しむ現象に苦悩していた［W6］。具体的には、①forward側ファイル（QC.1.trimmed.fastq）のsingle-endアセンブル結果が1 transcripts（107 bp）、②reverse側ファイル（QC.2.trimmed.fastq）のsingle-endアセンブル結果が423 transcripts（平均437 bp）、そして③paired-endのアセンブル結果が0 transcriptsであった（Rockhopper2 ver. 2.0.3）。

①

Rの基本的な利用法とパッケージのインストール

Bio-Linux 8にはR[31]がプレインストールされている。著者らの環境では、2015年4月にリリースされたR（ver. 3.2.0）が利用可能である。Biostringsなどいくつかの代表的なパッケージもプレインストールされているものの、マッピングからカウントデータ取得まで行えるQuasRを含む比較的最近のパッケージは、インストールから行う必要がある。ここでは、ゲストOS（Bio-Linux8）上でのRの基本的な利用法とQuasRパッケージのインストール法を示す。ホストOS（WindowsやMacintosh）上でのR

第5回原稿PDFのp195

この後は、①RパッケージQuasRによる乳酸菌ゲノムへのマッピングの話。そしてそれに関連したLinux環境でのRの利用の話に移行

として、FastQC によるクオリティチェックを行えばよい［W1-2］。著者らは、FastQC 実行結果ファイルの項目（Overrepresented sequences）を眺めて、トリム前に見えていた既知のアダプターやプライマー配列が、トリム後に正しく見えなくなっていることを確認して安心している［W1-3］。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

トランスクリプトームアセンブリ

ゲノムのアセンブリは、断片化されたゲノム配列由来リードをつなぎ合わせて、元のゲノム配列を再構築する作業である。この再構築に相当する英語がアセンブリ（assembly）であり、再構築を行うプログラムをアセンブラ（assembler）という。デノボ（*de novo*）という言葉が同時に用いられることが多いが、これは「最初から」と

か「一から」という意味である。このため、入力として（つまり他の情報を一切利用せずに）アセンブルする際には、*de novo* assembly という表現がなされる。トランスクリプトームアセンブリとは、アセンブル対象がゲノムではなく解析サンプル中で発現している全転写物（トランスクリプトーム）の場合を指す。RNA-seq データのみを入力として一からアセンブルする場合は、*de novo* transcriptome assembly などと呼ばれる。

このデータに関して結論からいえば、forward 側の 107 bp のリードファイル（SRR616268sub_1.fastq.gz → QC.1.trimmed.fastq）のうち、100–107 塩基付近に乳酸菌に由来しないものがトリムしきれずに多く残っている。これは、アセンブルやマッピングがうまくできない、という実害を被ることでわかる。計算時間がかかるため、できるだけ QC 段階で問題解決するという方針もあろう。しかし、やってみてはじめてわかることもある。以降の内容は、著者らが実際に行ったことを問題解決に至る思考回路とともに述べる。大まかに述べると、Rockhopper2 [18] によるトランスクリプトームアセンブリ、QuasR [19] による乳酸菌ゲノムへのマッピング、そして QC 再実行である。

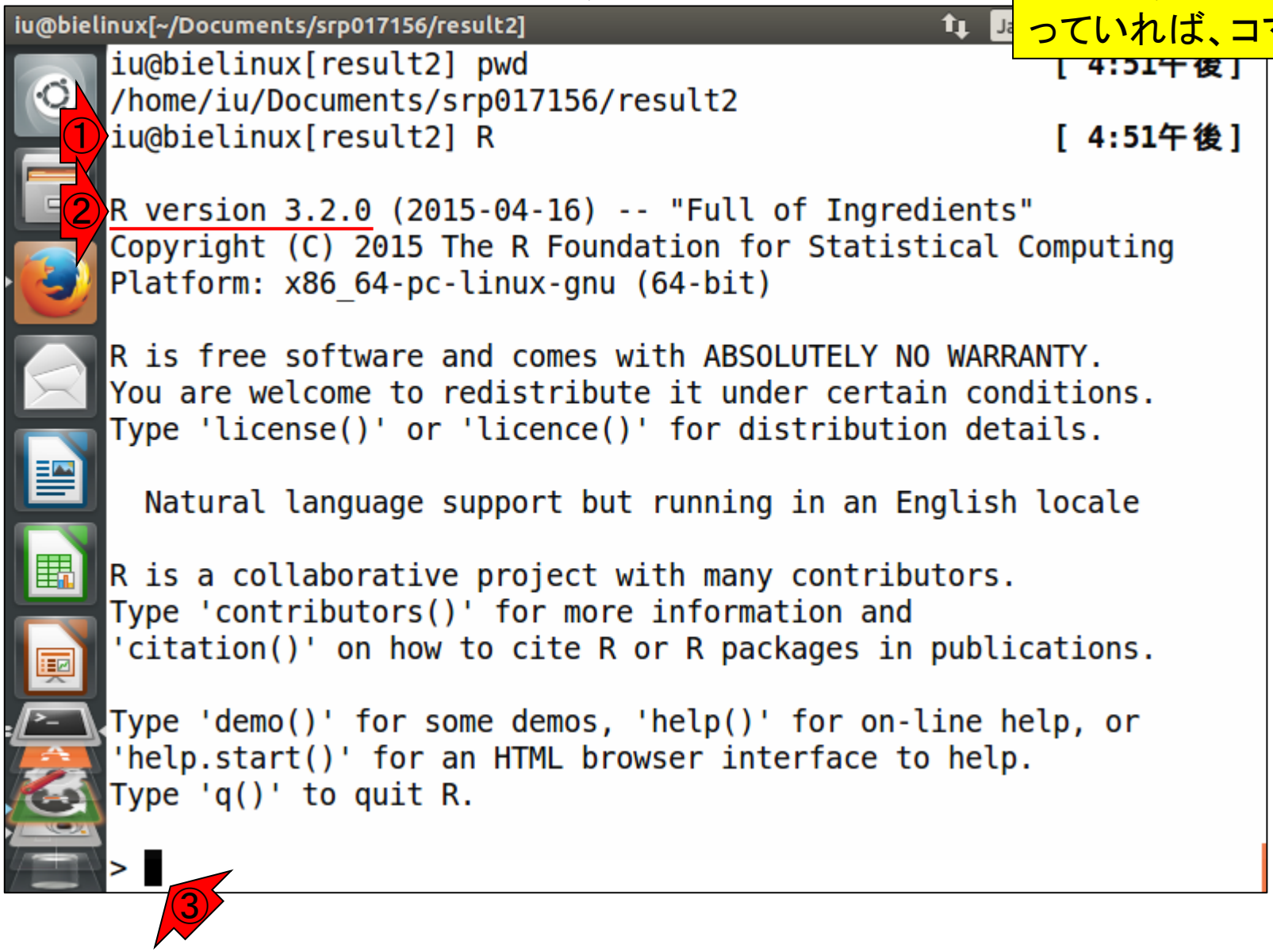
①

Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

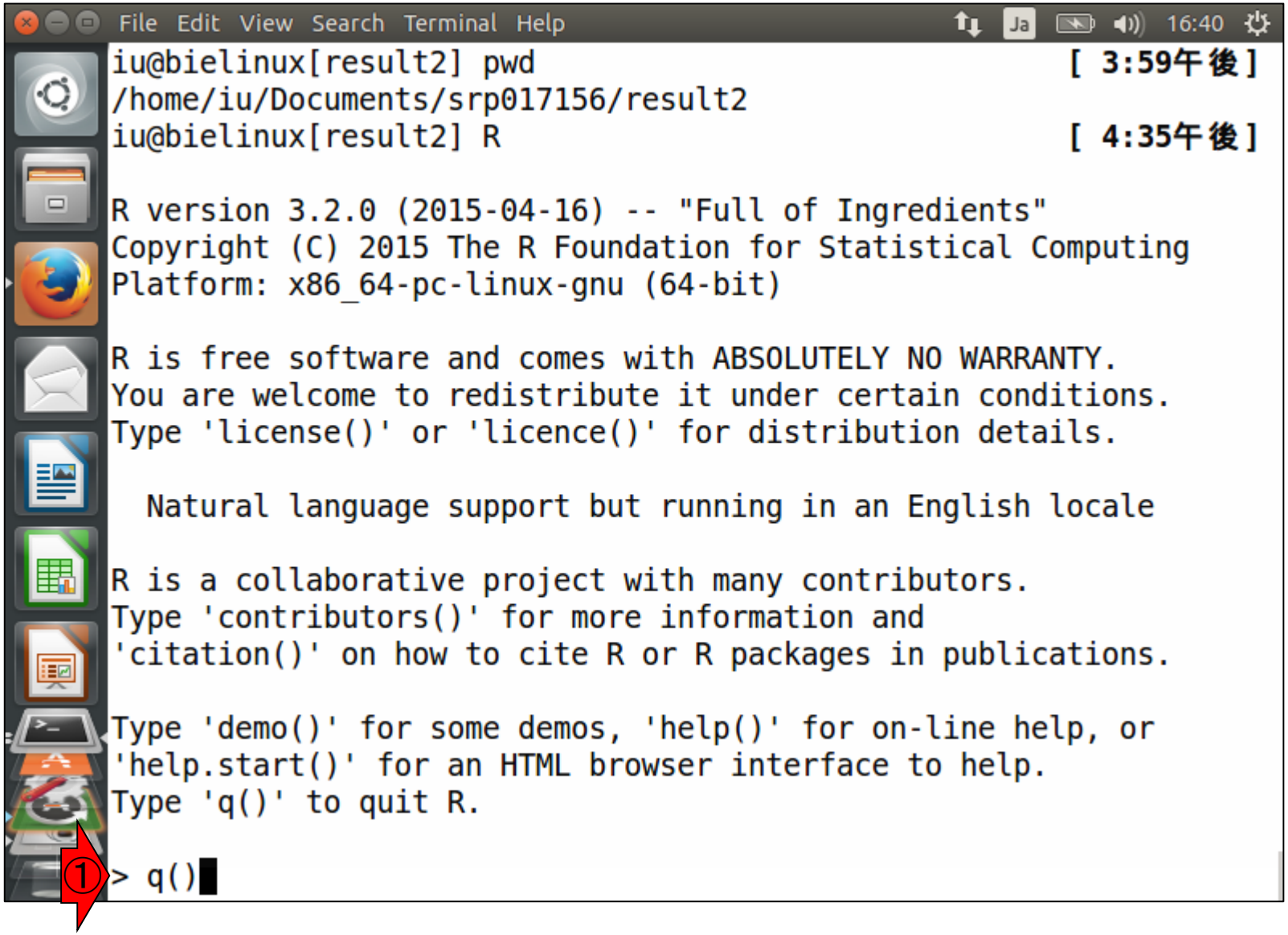
W7-1：Rの起動

①Rの起動は、「R」と打ってリターンキーを押すだけ。②Rのバージョンは3.2.0であることがわかる。③「>」となっていれば、コマンド入力待ち状態



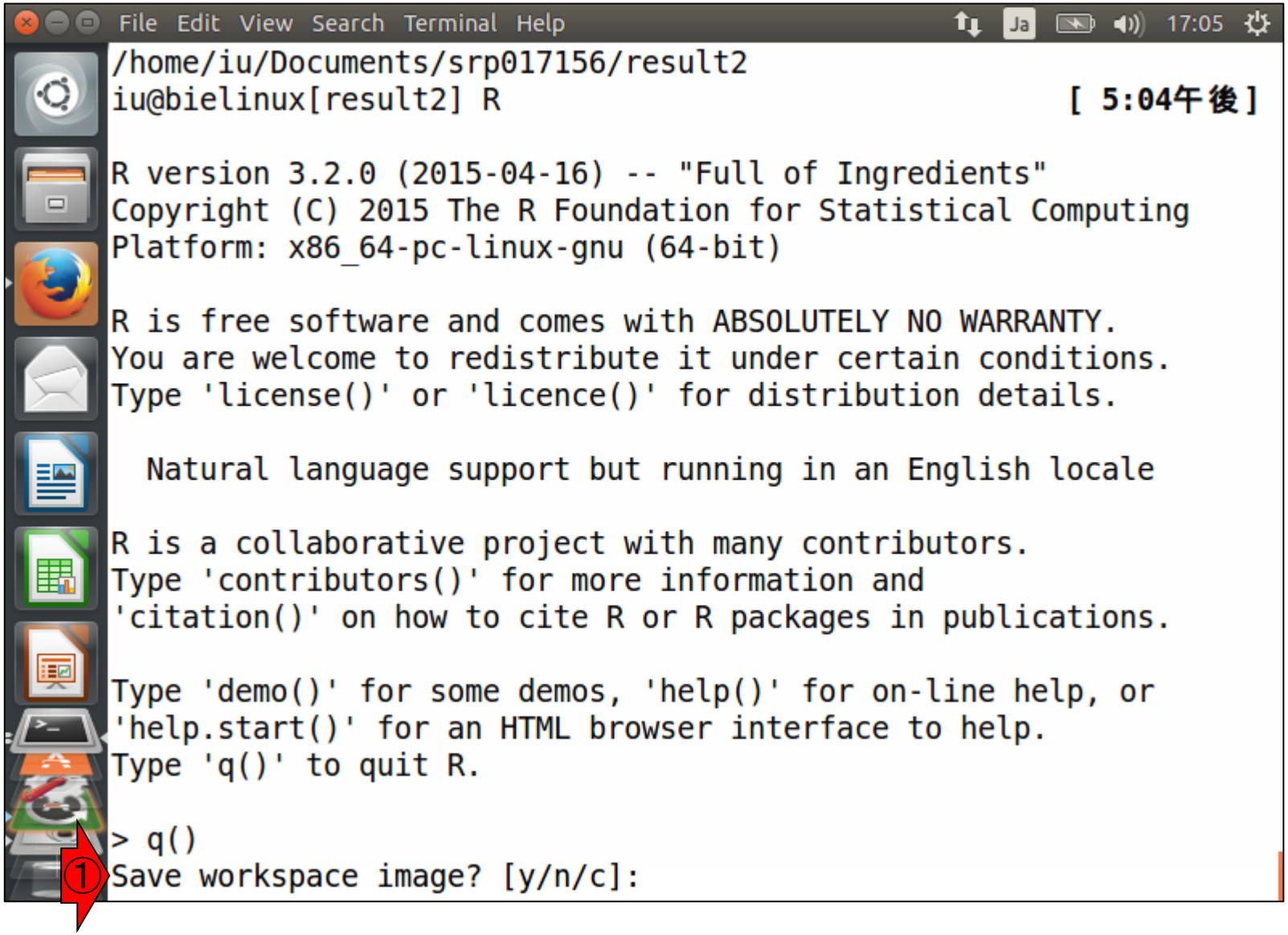
①Rの終了は「q()」

W7-1:Rの終了



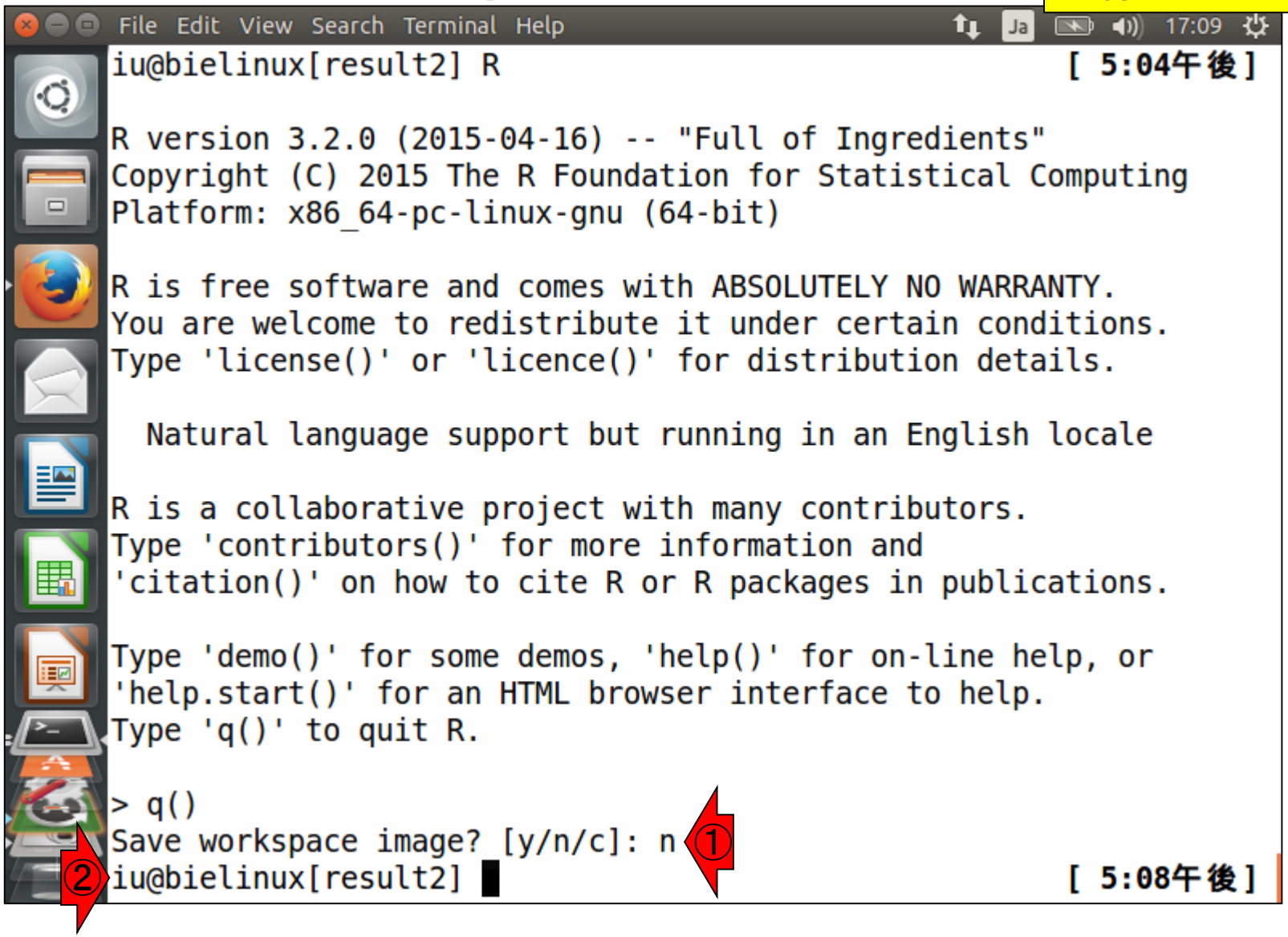
W7-1：Rの終了

①「Save workspace image?」と聞かれる。この意味がよくわからないうちは、Noに相当する「n」を打つ



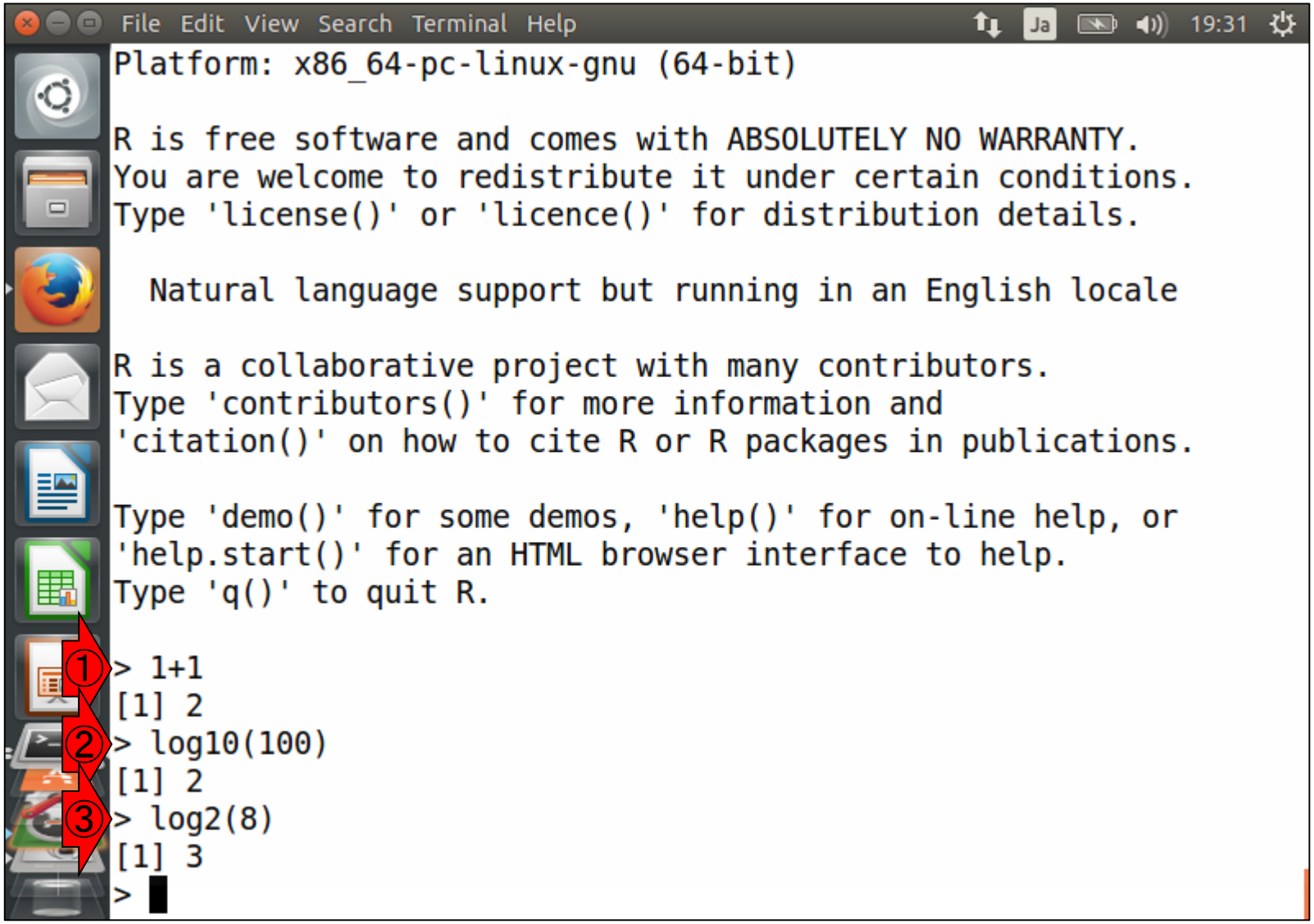
W7-1：Rの終了

①「n」と打ってリターンした直後の状態。②通常のBio-Linuxのコマンド入力待ち状態に戻ったことがわかる



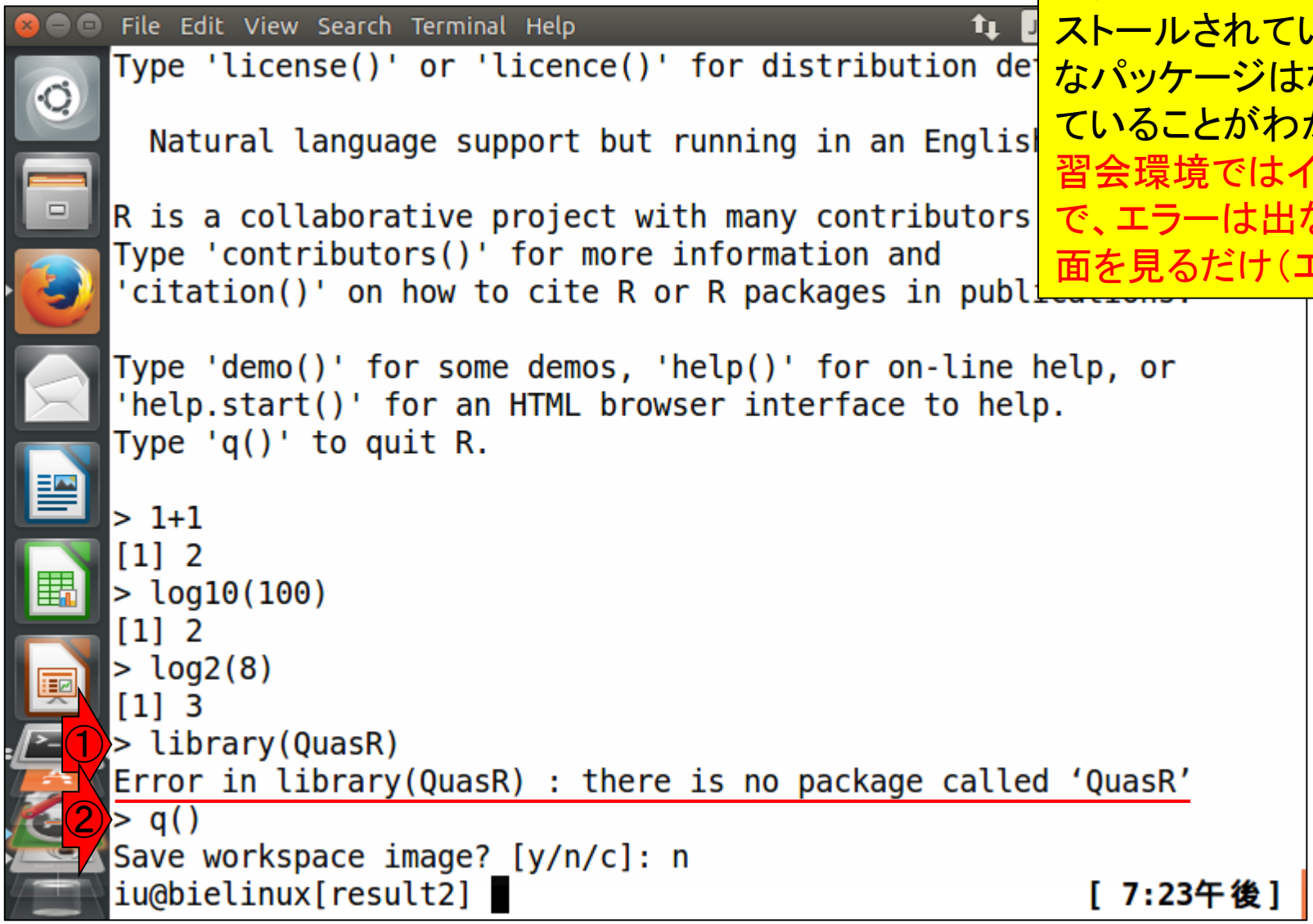
W7-2：基本的な利用法

もう一度Rを起動。簡単な数値計算ができます

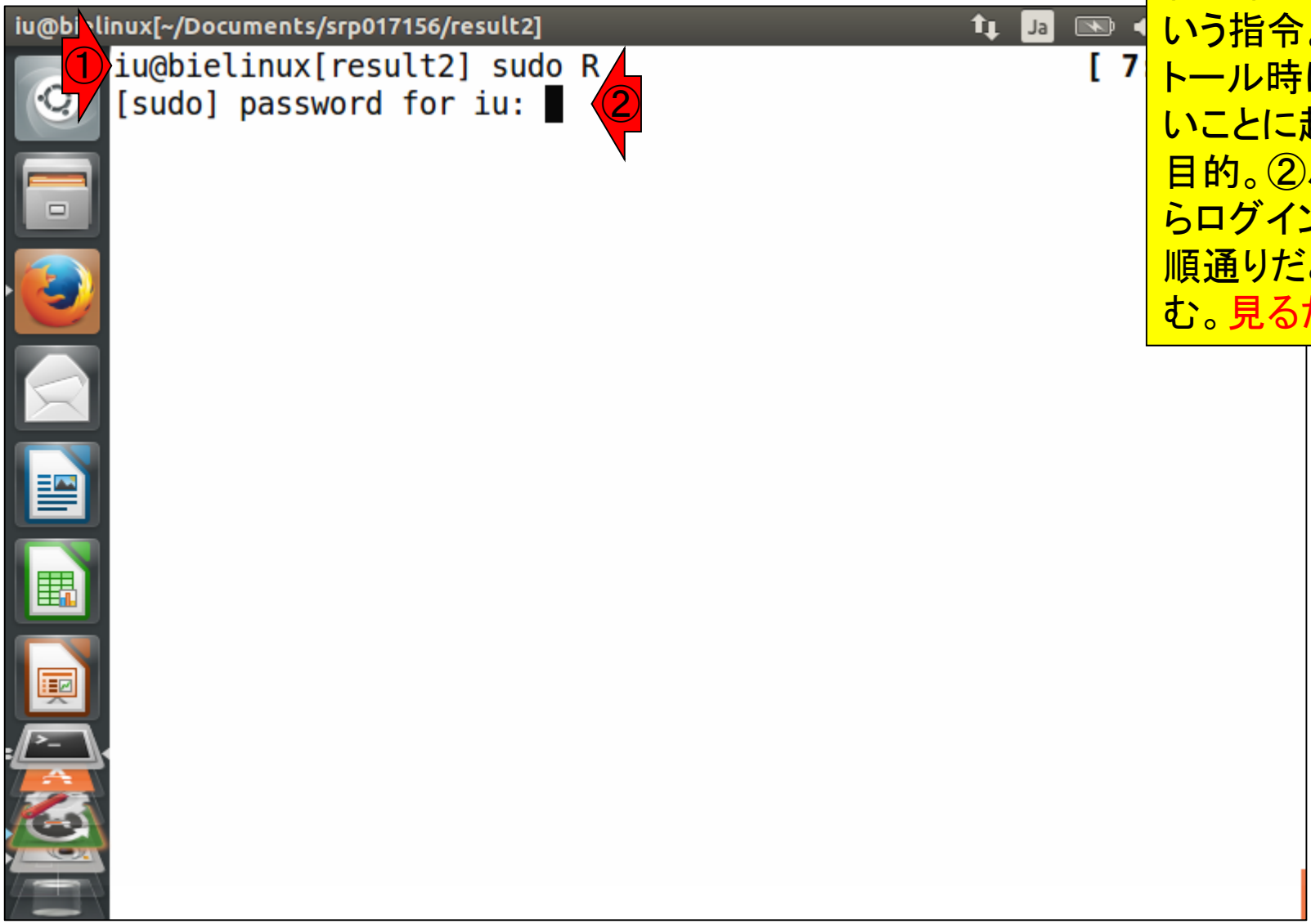


W7-3：パッケージのロード

①QuasRパッケージを利用したい場合はlibraryという関数を用いてロードする。Bio-Linux8にはまだQuasRがインストールされていないので、「そのようなパッケージはない」と文句を言われていることがわかる。②一旦終了。講習会環境ではインストール済みなので、エラーは出ない。このあたりは画面を見るだけ（エアーハンズオン）

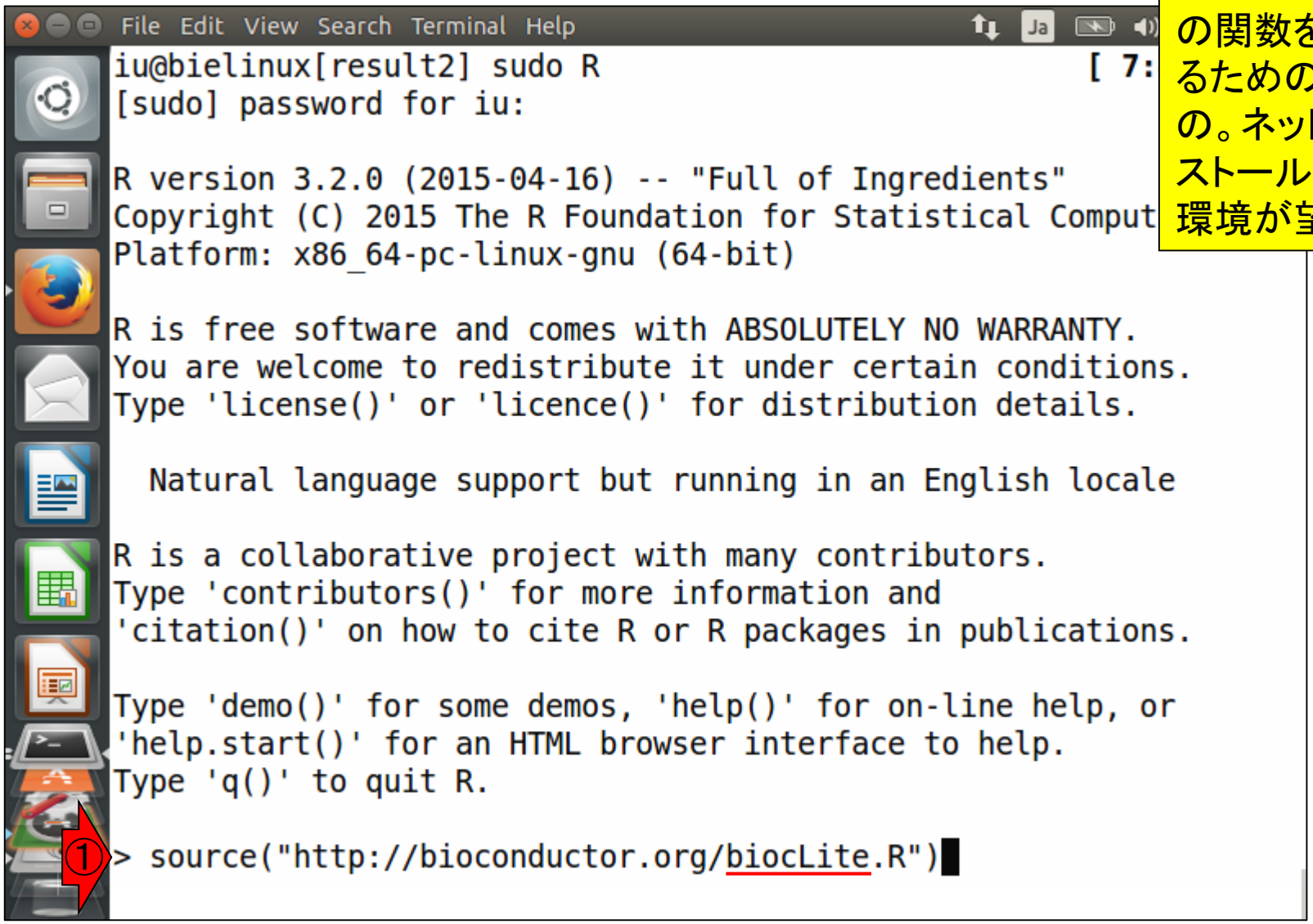


W7-4：パッケージインストール



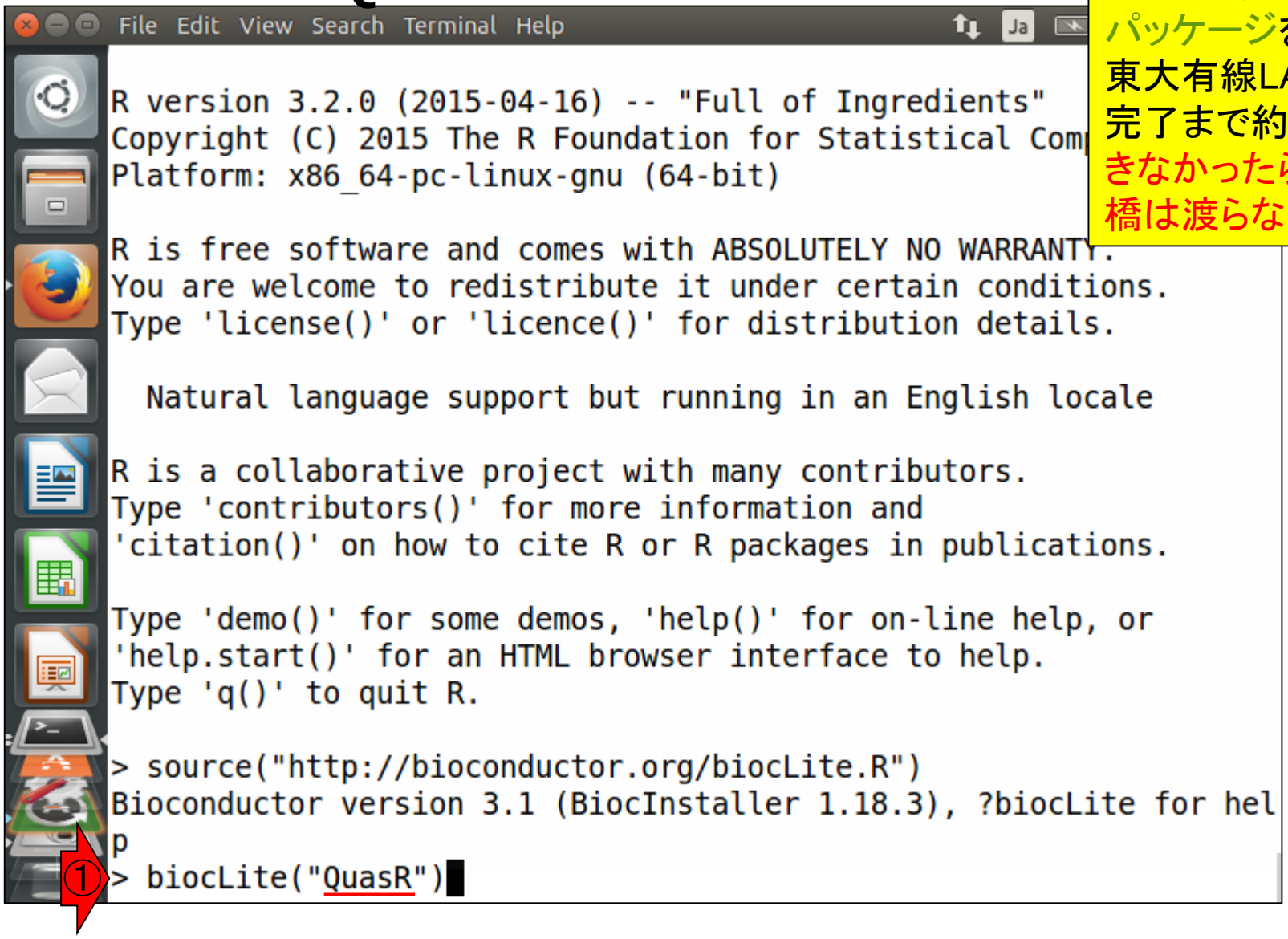
①作業ディレクトリはどこでもいいので「sudo R」。これはroot（管理者）権限でRを実行するという指令。パッケージのインストール時に書き込み権限がないことに起因するエラー回避が目的。②パスワードを聞かれたらログインパスワード（推奨手順通りだとpass1409）を打ち込む。見るだけ

W7-4：パッケージインストール



①「source("…")」を打ち込む。これは赤下線部分で示すbiocLiteというインストール用の関数を利用できるようにするためのおまじないのようなもの。ネットワーク経由でのインストール作業になる。有線LAN環境が望ましい。見るだけ

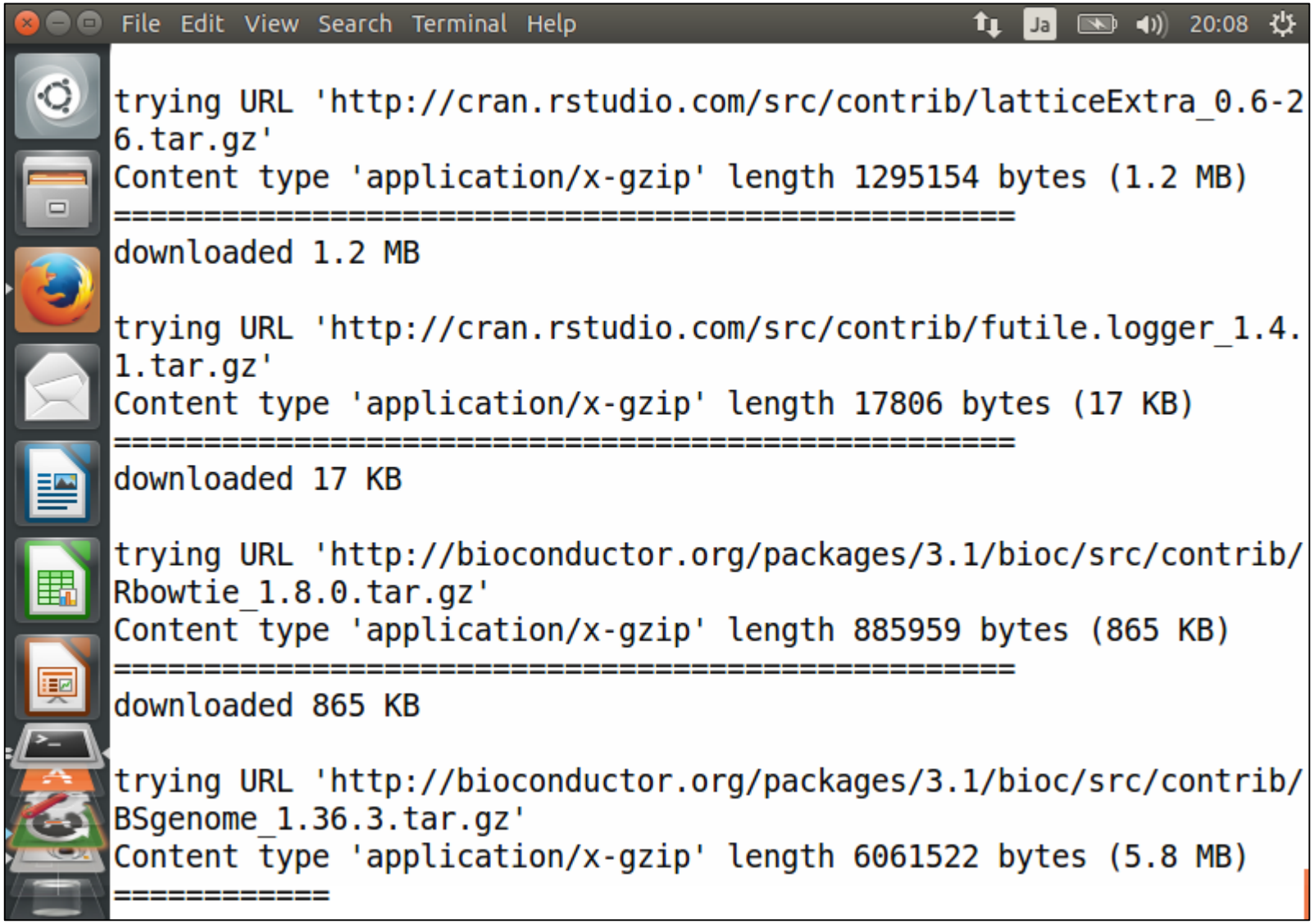
W7-4：QuasRインストール



①biocLite関数を用いてQuasRをインストール。赤下線部分を変えることで、同じノリで他の任意のパッケージをインストール可能。東大有線LAN環境でインストール完了まで約20分。講習会当日できなかったら後々困るので危ない橋は渡らないのです。見るだけ

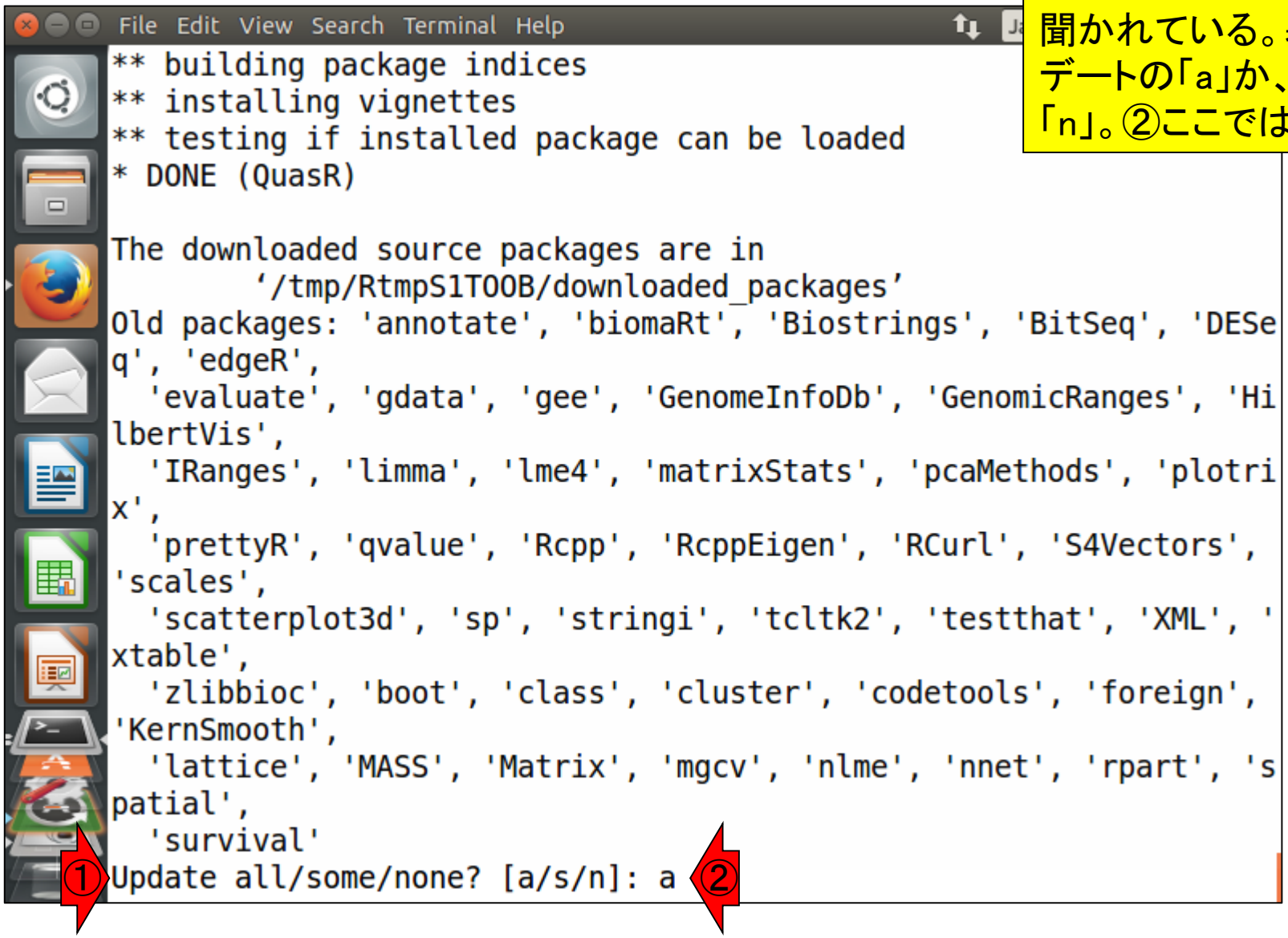
数秒後の状態

W7-5：途中経過1



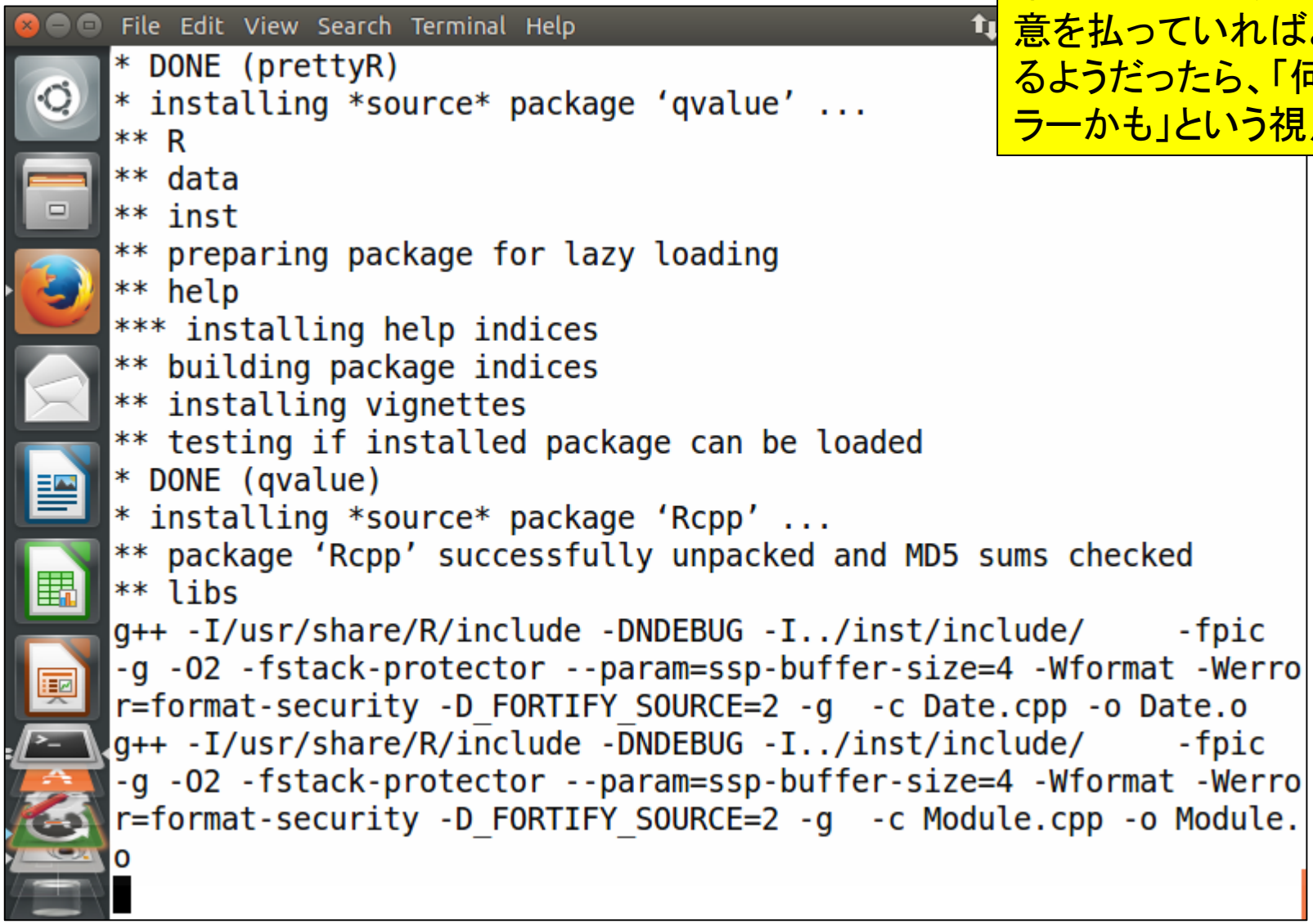
W7-5：途中経過2

①リターンキーを押してから約5分後にこのような状態になる。古いパッケージのアップデートをするかどうかを聞かれている。基本はすべてアップデートの「a」か、アップデートしないの「n」。②ここでは「a」と打ってリターン



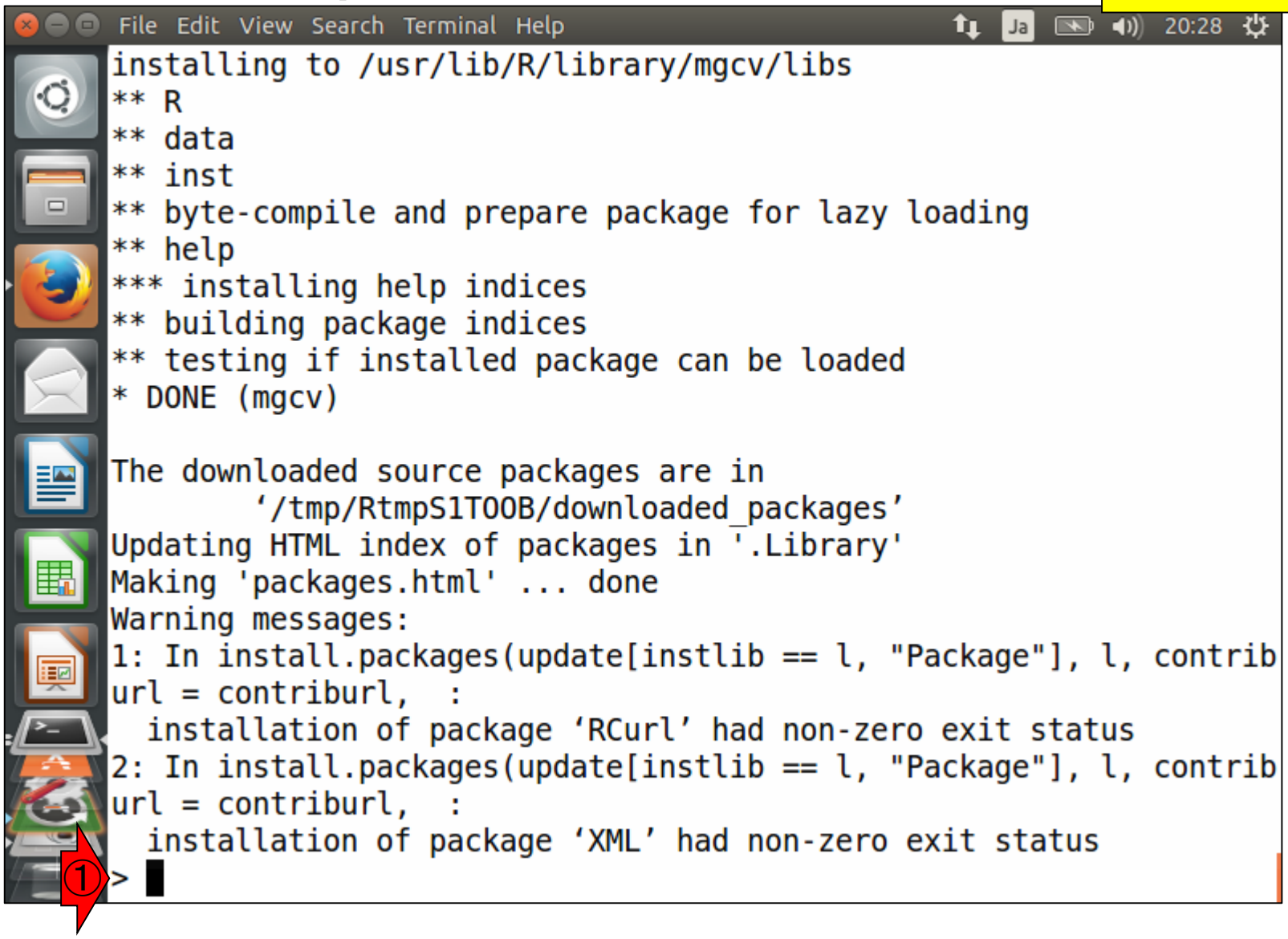
W7-5：途中経過3

特にエラーを吐くことなく順調にインストールが進んでいるようだ。この間は、基本的に画面が流れているかどうかに注意を払っていればよい。もし止まっているようだったら、「何か聞かれているかエラーかも」という視点でメッセージを見る



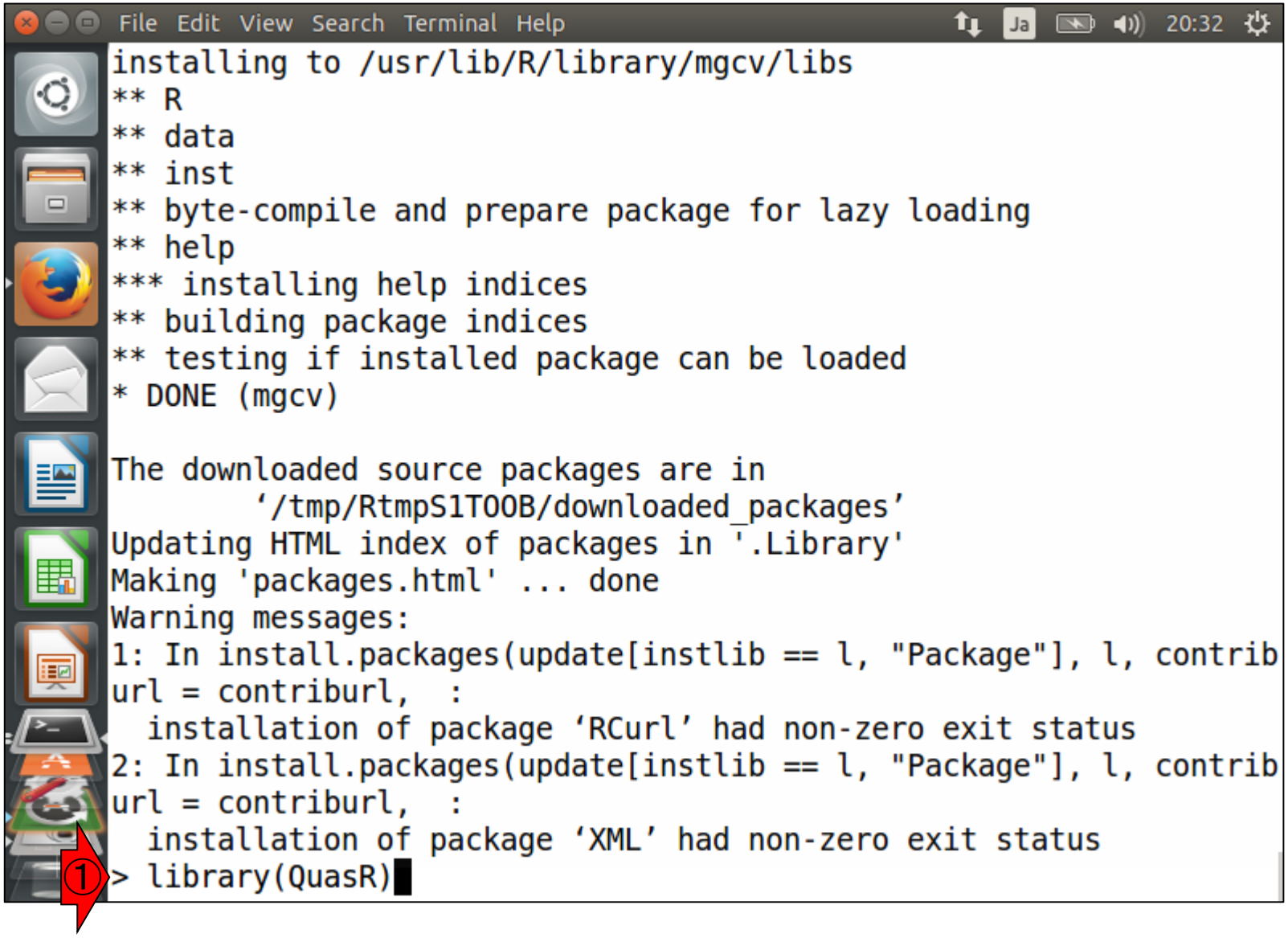
W7-5：終了後の状態

①コマンド入力待ち状態になれば基本的にOK。パッと見、エラーメッセージが出ていないようだ



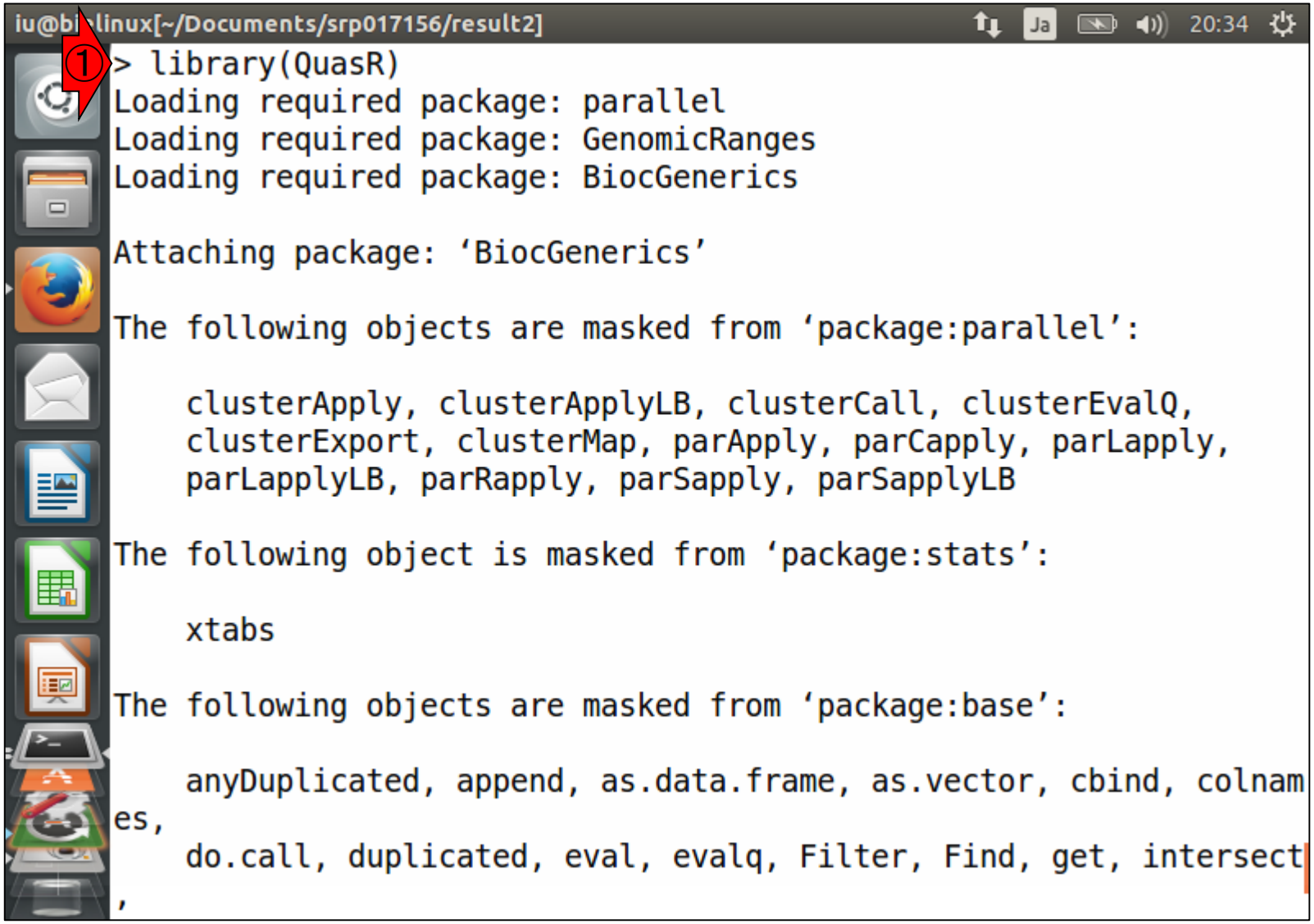
W7-6：インストール確認

①「library(QuasR)」を実行
。ここからは実際に行う



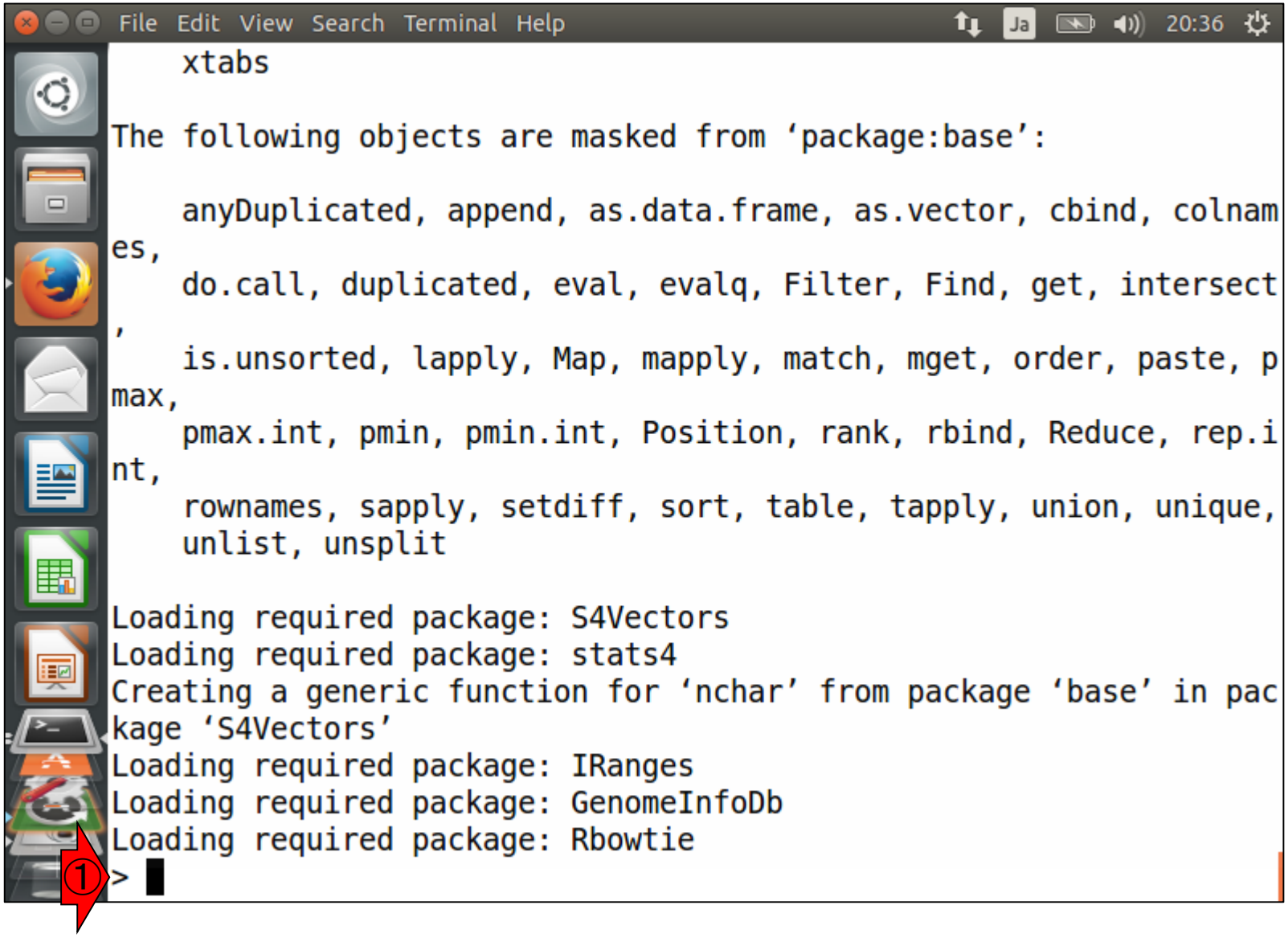
①リターンキーを押した最初のほうの画面

W7-6：インストール確認



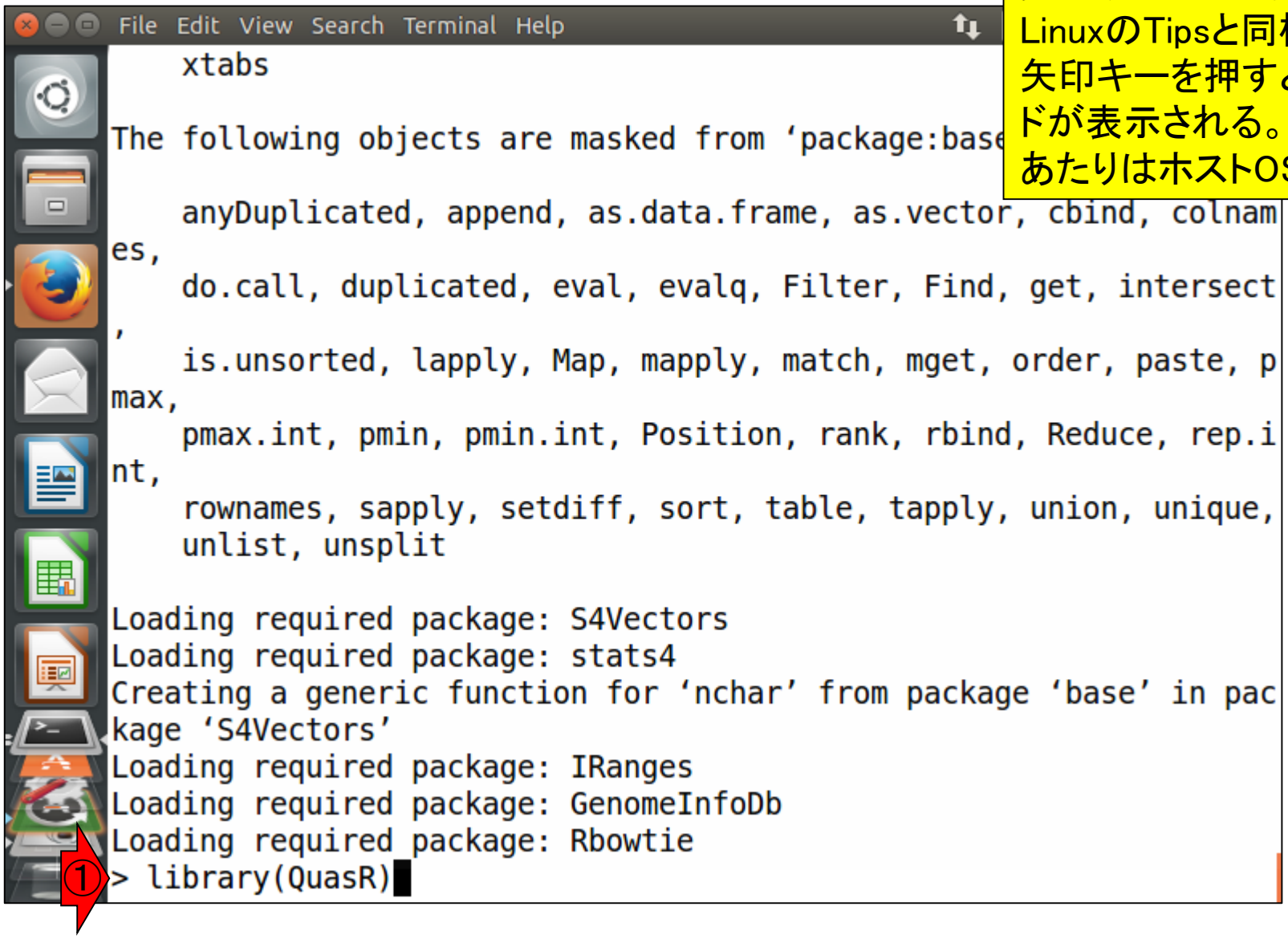
W7-6：インストール確認

リターンキーを押した最後のほうの画面。
特にエラーメッセージは出ていないようだ



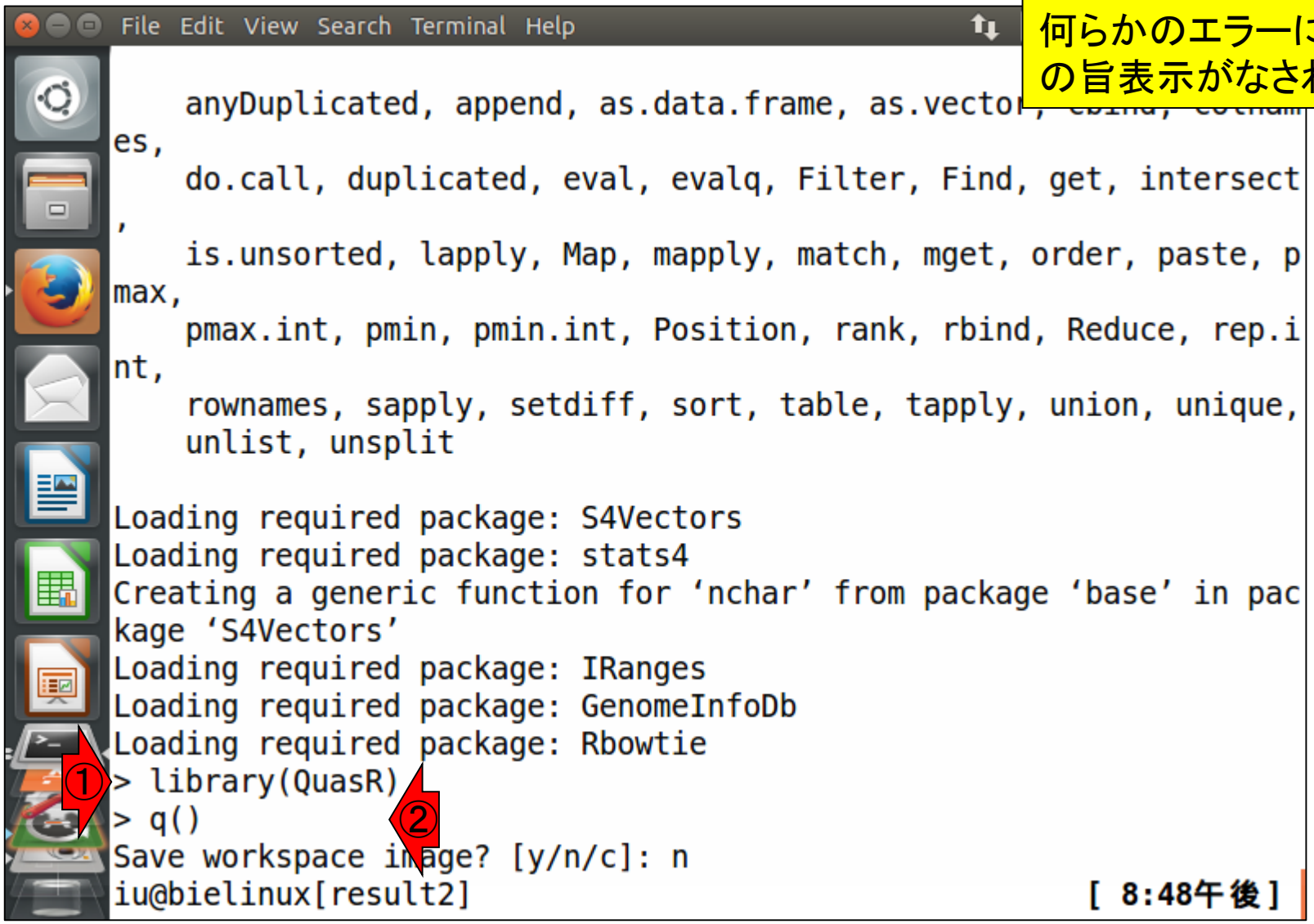
W7-6：インストール確認

画面がばーっと流れてエラーの確認がしづらいときは、もう一度同じコマンドを実行するとよい。このとき、一般的なLinuxのTipsと同様に、キーボードの上矢印キーを押すと直前に打ったコマンドが表示される。有効利用すべし。このあたりはホストOS上のRと同じ使用感



W7-6：インストール確認

①2回目は、特に何も表示されない。このような場合は、QuasRパッケージのロードに成功していることを意味する。何らかのエラーに遭遇していれば、その旨表示がなされる。②一旦Rを終了



W7-7：QuasRウェブページ

BioconductorのQuasRウェブページ。QuasRパッケージのインストール手順と実際に行ったこととの対応関係がよくわかるでしょう。スライドを見るだけ

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Search:

Home | Install | Help | Developers | About

Home » Bioconductor 3.1 » Software Packages » QuasR

QuasR

platforms some | downloads top 5% | posts 2 / 2 / 2 / 0 | in Bioc 2.5 years
build ok | commits 4.00 | test coverage 21%

Quantify and Annotate Short Reads in R

Bioconductor version: Release (3.1)

This package provides a framework for the quantification and analysis of Short Reads. It covers a complete workflow starting from raw sequence reads, over creation of alignments and quality control plots, to the quantification of genomic regions of interest.

Author: Anita Lerch, Dimos Gaiditzis and Michael Stadler

Maintainer: Michael Stadler <michael.stadler at fmi.ch>

Citation (from within R, enter `citation("QuasR")`):

Gaidatzis D, Lerch A, Hahne F and Stadler MB (2015). "QuasR: Quantification and annotation of short reads in R." *Bioinformatics*, **31**(7), pp. 1130–1132. http://dx.doi.org/10.1093/bioinformatics/btu781, PMID:25417205.

Langmead B, Trapnell C, Pop M and Salzberg SL (2009). "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome Biology*, **10**(3), pp. R25. http://dx.doi.org/10.1186/gb-2009-10-3-r25, PMID:19261174.

Au KF, Jiang H, Lin L, Xing Y and Wong WH (2010). "Detection of splice junctions from paired-end RNA-seq data by SpliceMap." *Nucleic Acids Research*, **38**(14), pp. 4570–4578. http://dx.doi.org/10.1093/nar/gkq211, PMID:20371516.

Installation

To install this package, start R and enter:

```
## try http if https is not available
source("https://bioconductor.org/biocLite.R")
biocLite("QuasR")
```

Documentation

To view documentation for the version of this package installed in your system, start R and enter:

```
browseVignettes("QuasR")
```

Documentation »

Bioconductor

- Package vignettes and manuals.
- Workflows for learning and use.
- Course and conference material.
- Videos.
- Community resources and tutorials.

R / CRAN packages and documentation

Support »

Please read the posting guide. Post questions about Bioconductor to one of the following locations:

- Support site - for questions about Bioconductor packages
- Bioc-devel mailing list - for package developers

To install this package, start R and enter:

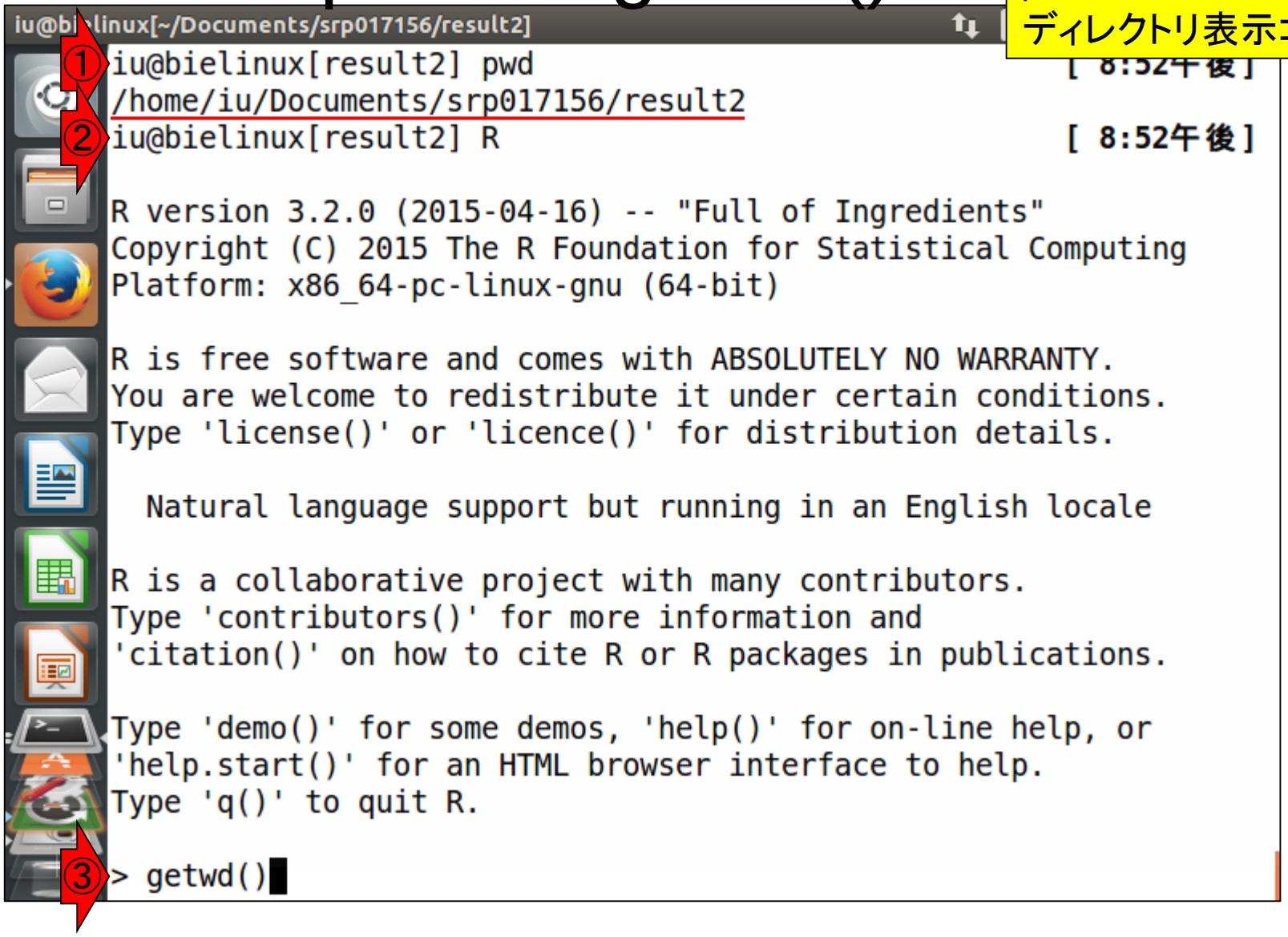
```
## try http if https is not available
source("https://bioconductor.org/biocLite.R")
biocLite("QuasR")
```

Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

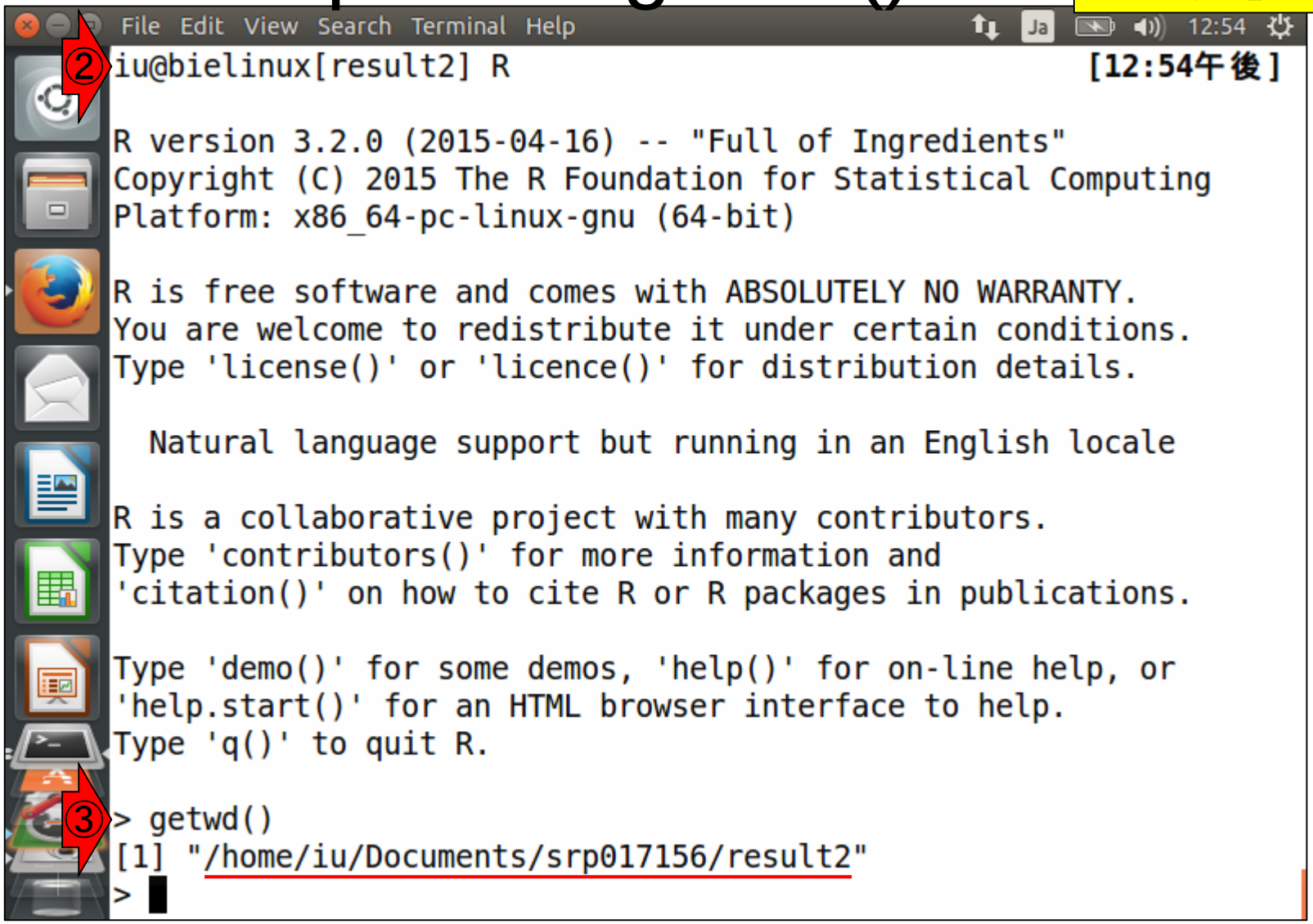
W8-1：pwd ⇔ getwd()

①現在の作業ディレクトリは赤下線で示したところ。②Rを起動。③Linuxのpwdコマンドに対応するR上での作業ディレクトリ表示コマンドは「getwd()」



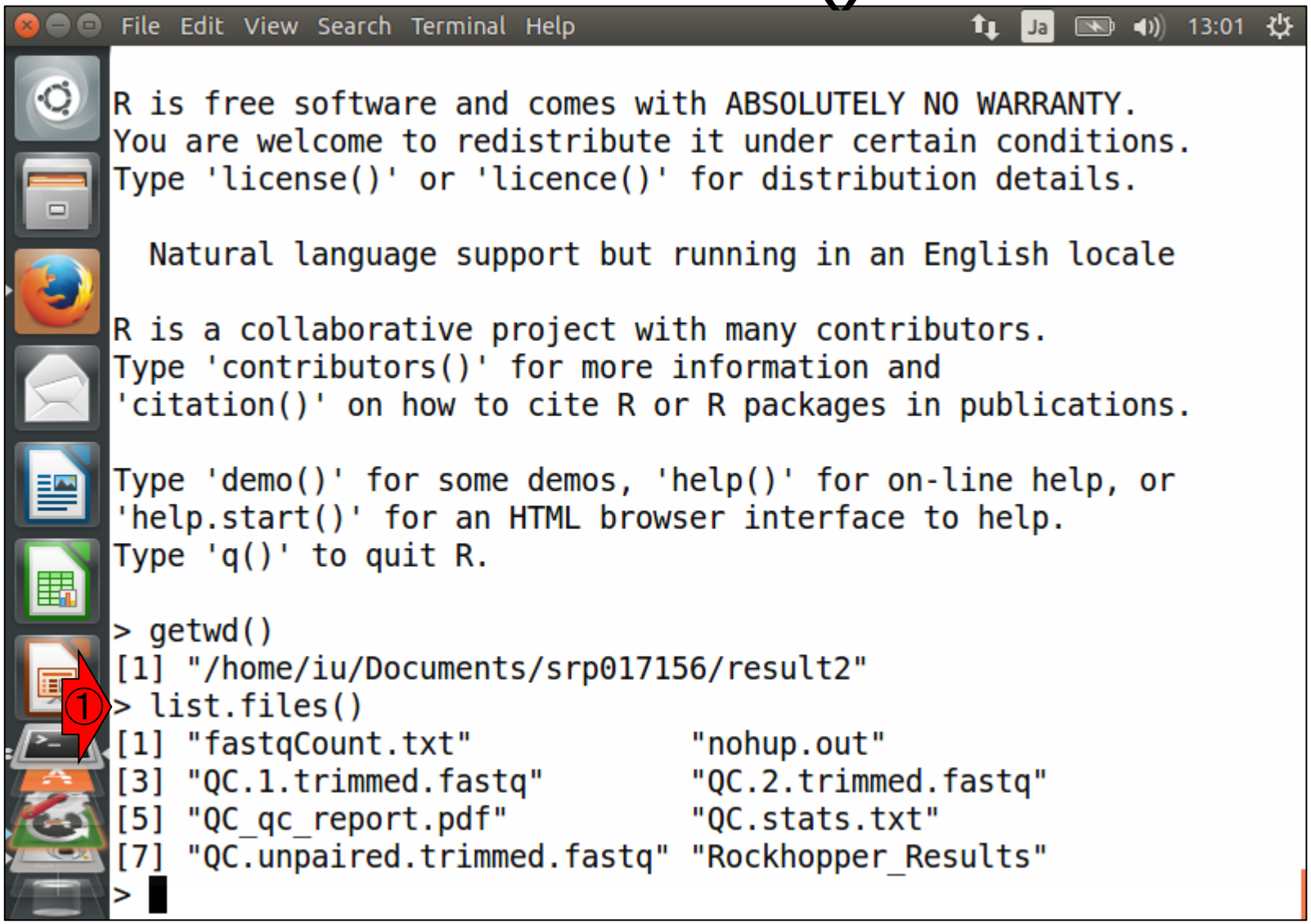
W8-1：pwd ⇔ getwd()

ホストOS（WinやMac）上のR GUI版との違いは、起動時の作業ディレクトリが、Rを起動した場所になる点



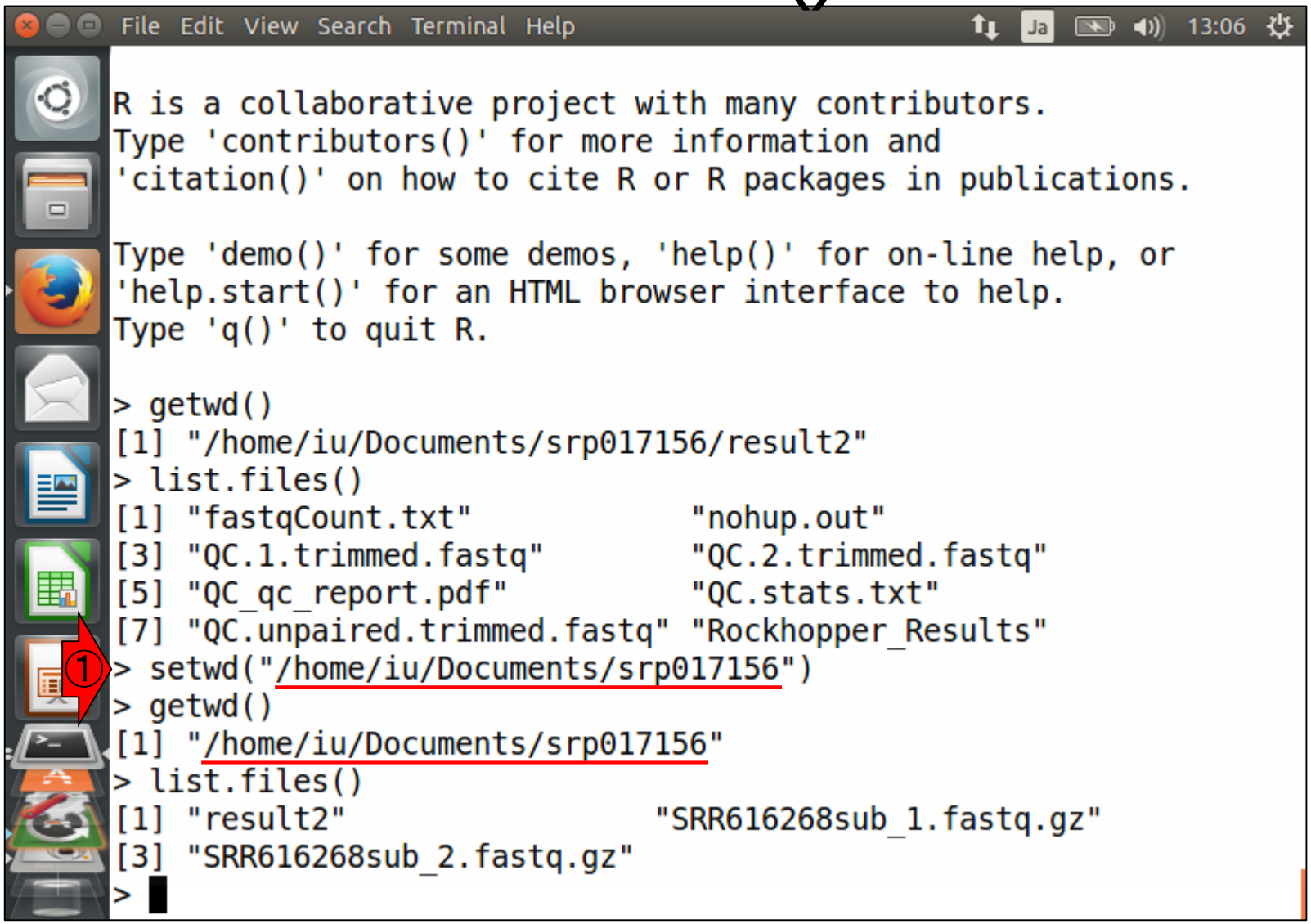
W8-2 : ls ⇔ list.files()

①LinuxのlsコマンドにRコマンドは「list.files()」



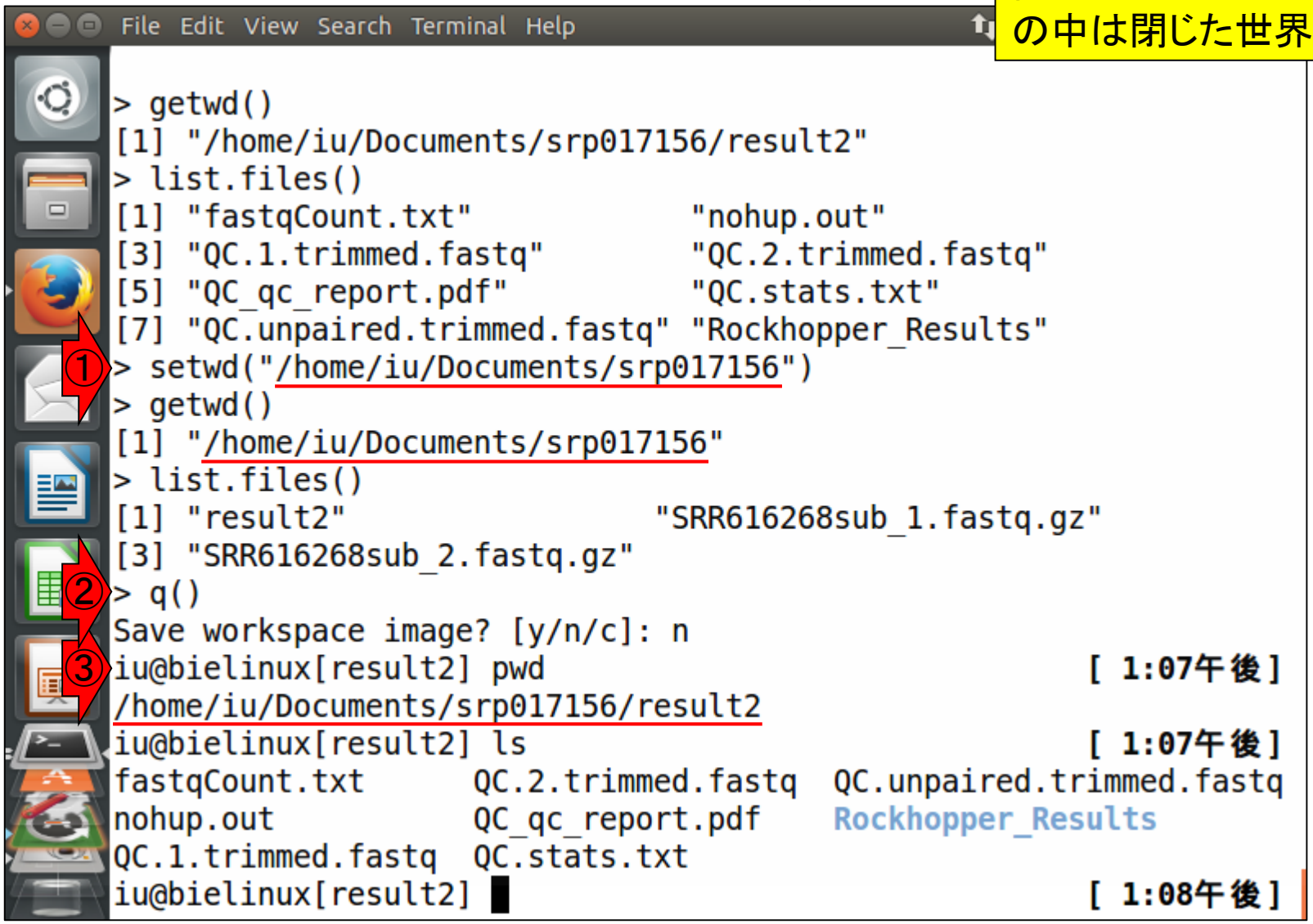
W8-3：cd ⇔ setwd()

①Linuxのcdコマンドに対応するRコマンドは「setwd()」



W8-4：Rは閉じた世界

①作業ディレクトリ変更後に②Rを終了。Linuxの世界に戻ったのち③pwd。R起動前のディレクトリと同じ。このことから、Rの中は閉じた世界であることがわかる

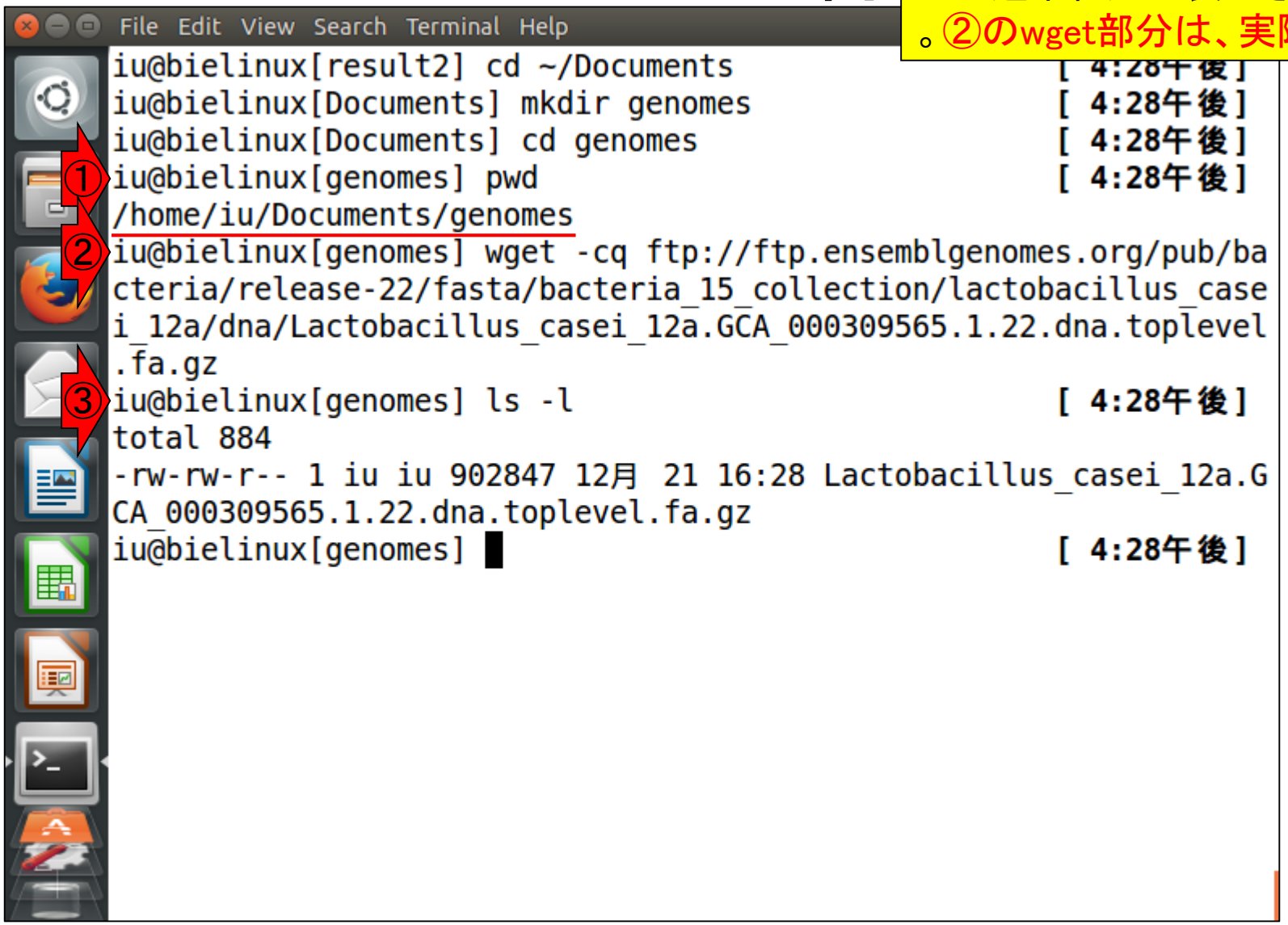


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、**W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）**
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

W9-1：ゲノム配列取得

①作業ディレクトリは「~/Documents/genomes」。②wget実行時にqオプションをつけているので途中経過が表示されなくてスッキリ。③ls。②のwget部分は、実際には行わないで!



W9-1：ゲノム配列取得

①普通にコピペしても行頭に#をつけてコメントアウトしているので、「#wgetというコマンドはない!」と言われて実行されません。講習会では「~/Desktop/backup」フォルダ上にダウンロード済みのファイルを置いてあります。それゆえ、実際には①wgetの代わりに②をコピペ実行

Rでゲノム解析(Linux版)

- Lactobacillus casei 12A (Taxonomy ID: 1051650): Broadbent et al.,
- Ensembl Genomes (Cunningham et al., Nucleic Acids Res., 2015)の
- ゲノム配列取得[W9-1]
  Ensembl release 22の乳酸菌(Lactobacillus casei 12A)ゲノム配列を取得。

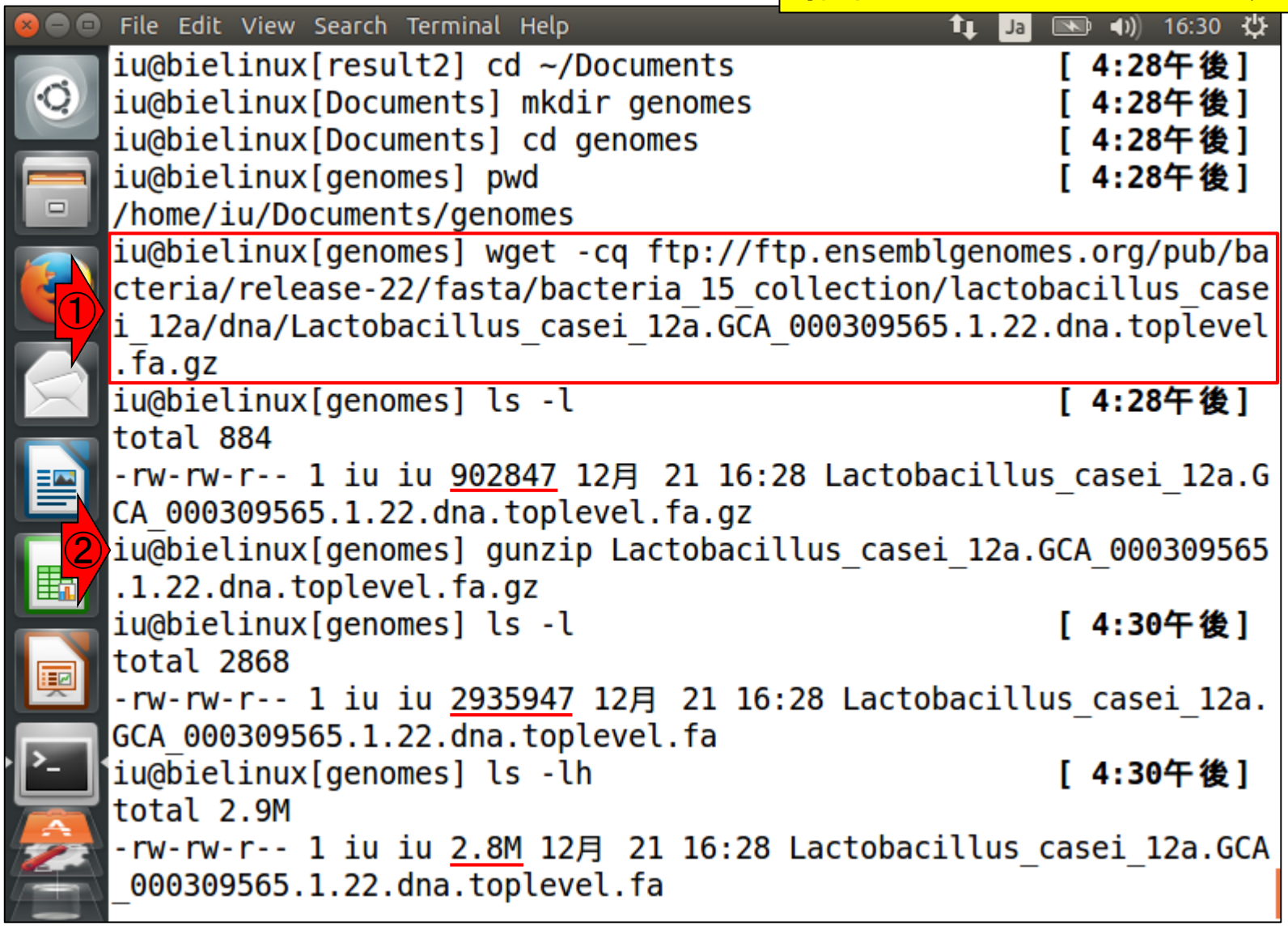
```
cd ~/Documents
mkdir genomes
cd genomes
pwd
#wget -cq ftp://ftp.ensemblgenomes.org/pub/bacteria/release-22/fasta/bacteria_15_collec
cp ~/Desktop/backup/Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa.gz .
ls -l

gunzip Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa.gz
ls -l
ls -lh
```

①

②

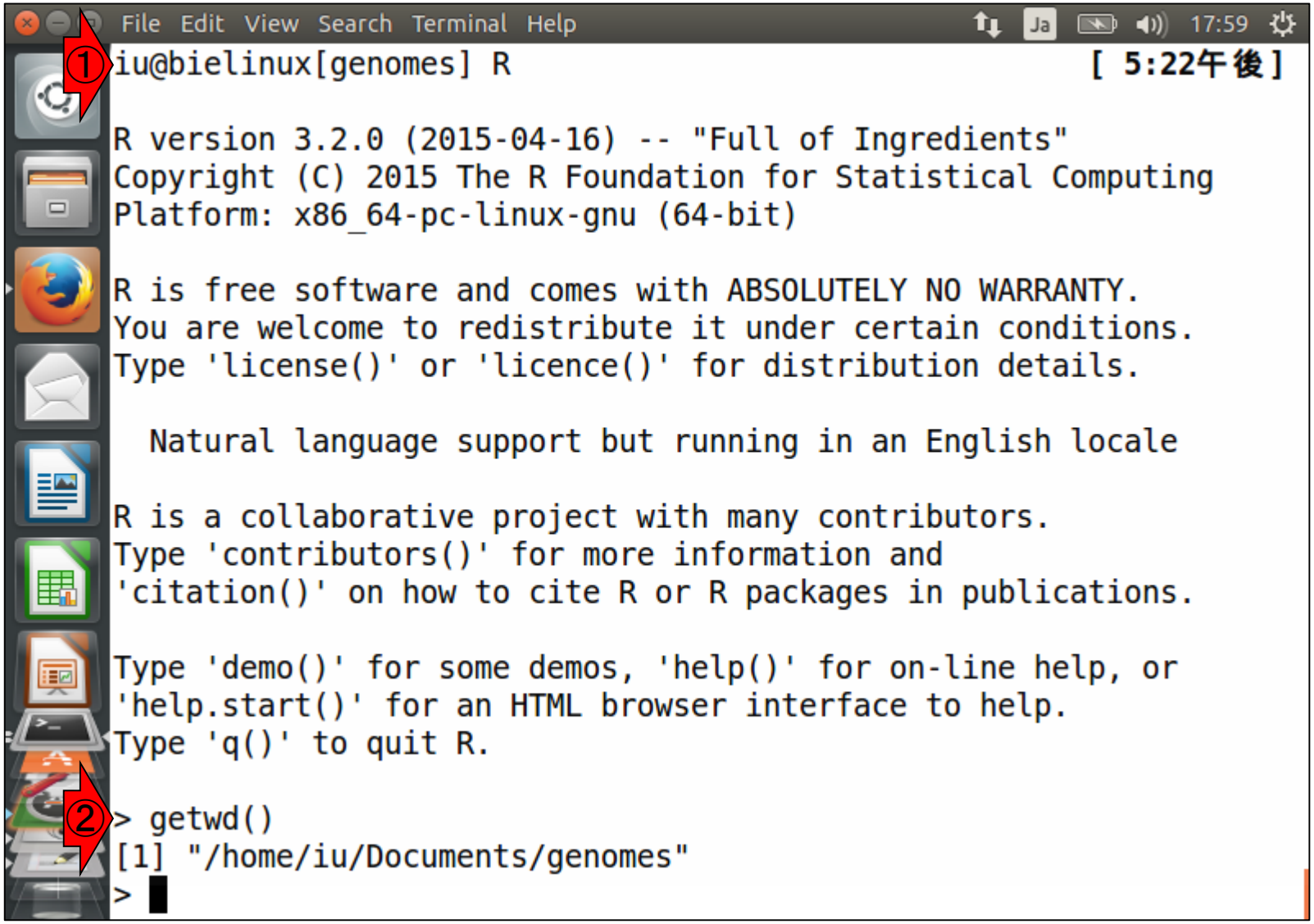
W9-1：ゲノム配列取

つまり、①の部分が「cp ~/Desktop/backup/Lac* .」になるということです。②gunzipでgzファイルを解凍。解凍後のファイルサイズは2,935,945 bytes（約2.8MB）



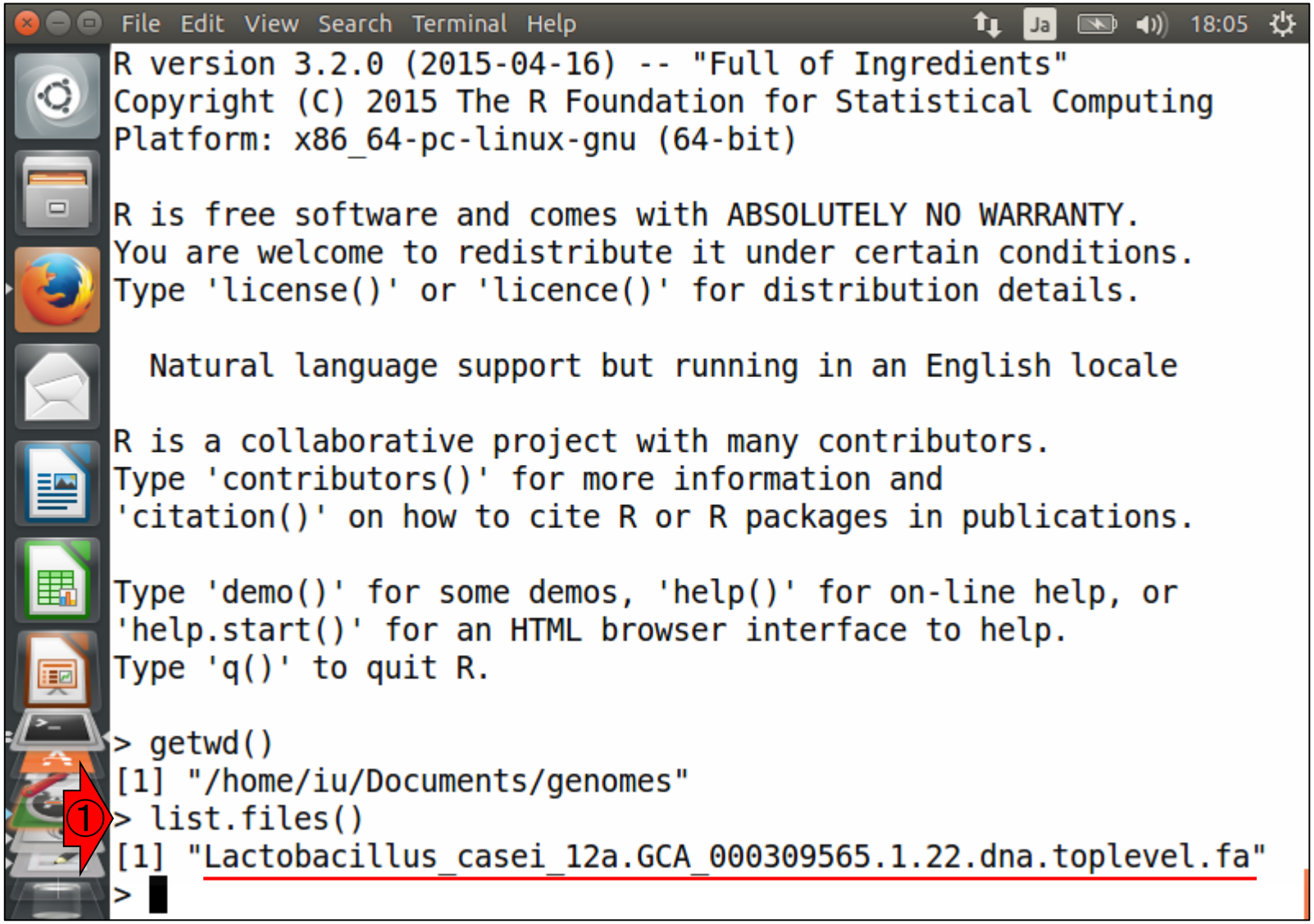
W9-2：Rを起動

「~/Documents/genomes」上で、①Rを起動。②getwd()はただの確認



①list.files()で入力ファイルの存在確認

W9-3：入力ファイルの確認



W9-4：コピペ用コード

入出力の関係。①入力ファイル名、②出力ファイル名

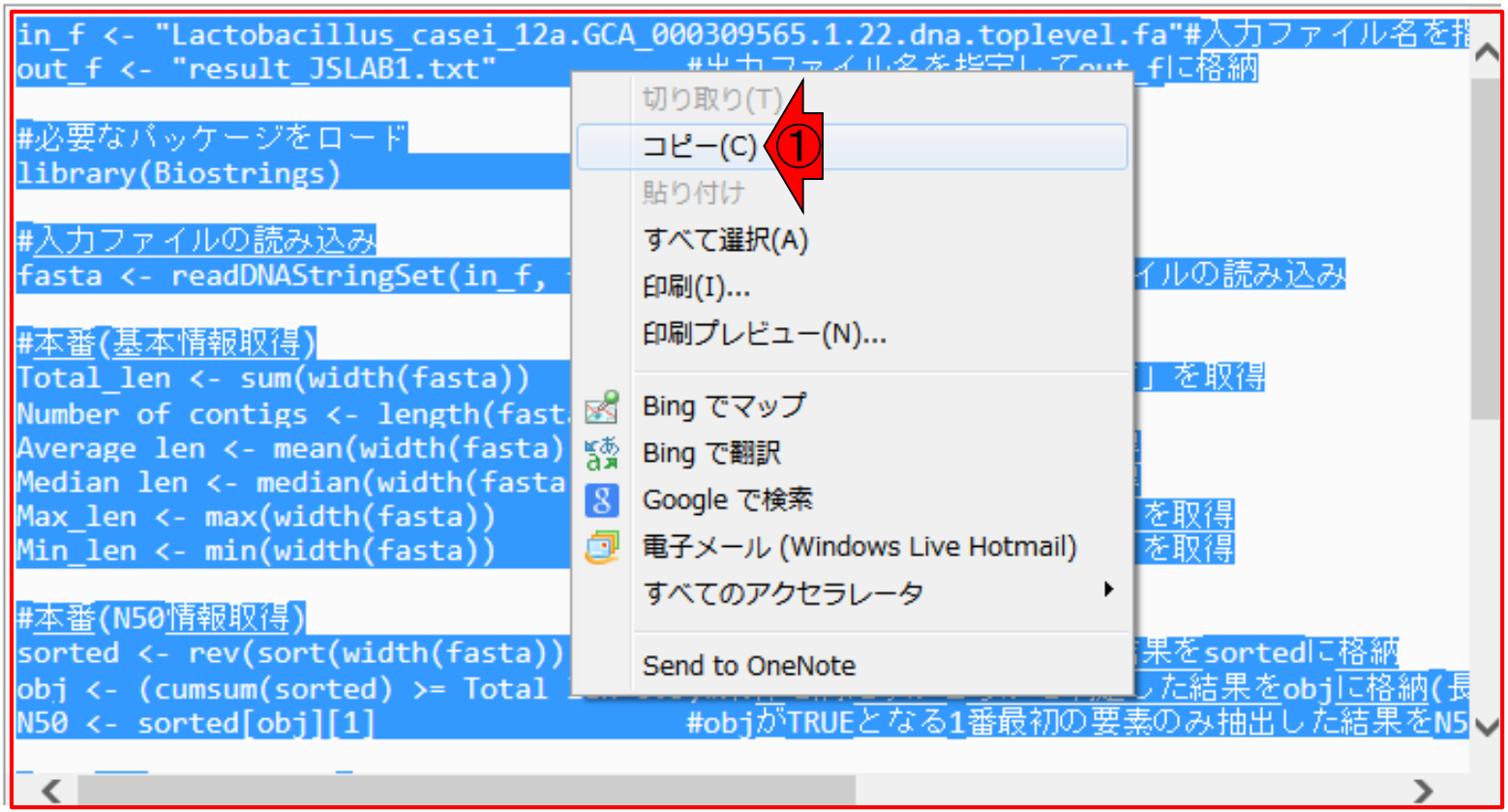
- 連載第1回の「Rでゲノム解析」と同じコード。
  このウェブページ中の「イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得」の5.と基本的に同じ。
  以下の内容はJSLAB5_1.Rの中身と同じです。

```
in_f <- "Lactobacillus_casei_12a.GCA_000309565.1.22.dna.toplevel.fa"  #入力ファイル名を
out_f <- "result_JSLAB1.txt"                #出力ファイル名を指定してout_fに格納

# 必要なパッケージをロード
library(Biostrings)                         #パッケージの読み込み

# 入力ファイルの読み込み
fasta <- readDNAStringSet(in_f, format="fasta")#in_fで指定したファイルの読み込み

# 本番(基本情報取得)
Total_len <- sum(width(fasta))              #配列の「トータルの長さ」を取得
Number_of_contigs <- length(fasta)          #「配列数」を取得
Average_len <- mean(width(fasta))           #配列の「平均長」を取得
Median_len <- median(width(fasta))          #配列の「中央値」を取得
Max_len <- max(width(fasta))                #配列の長さの「最大値」を取得
Min_len <- min(width(fasta))                #配列の長さの「最小値」を取得

# 本番(N50情報取得)
sorted <- rev(sort(width(fasta)))           #長さ情報を降順にソートした結果をsortedに格納
obj <- (cumsum(sorted) >= Total_len*0.5)#条件を満たすかどうかを判定した結果をobjに格納(
N50 <- sorted[obj][1]                       #objがTRUEとなる1番最初の要素のみ抽出した結果をN
```

W9-5：コピペ

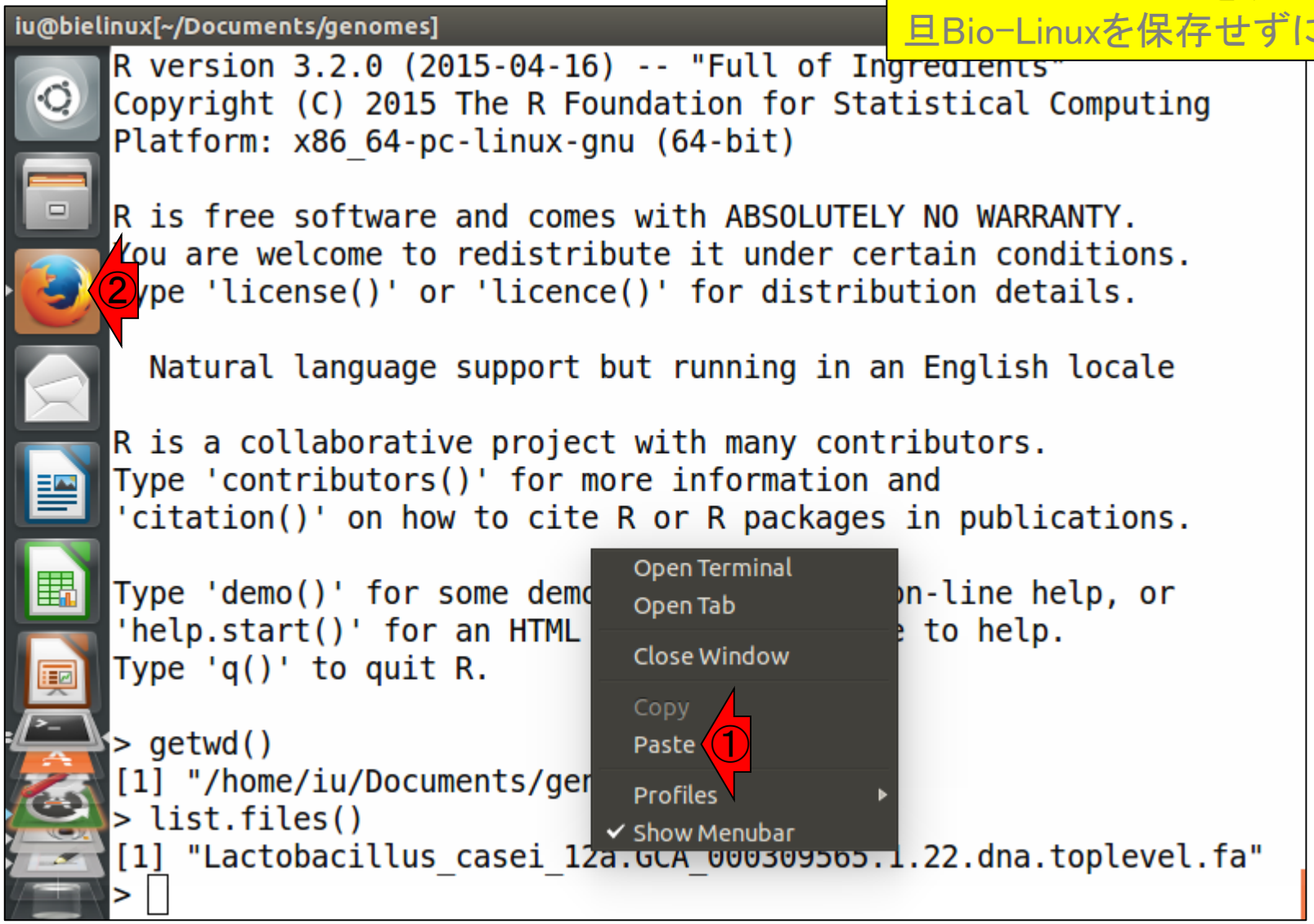
赤枠内（つまりコード全部）を全選択して①コピー

- 連載第1回の「Rでゲノム解析」と同じコード。
  このウェブページ中の「イントロ | NGS | 読み込み | FASTA形式 | 基本情報を取得」の5.と基本的に同じ。
  以下の内容はJSLAB5_1.Rの中身と同じです。



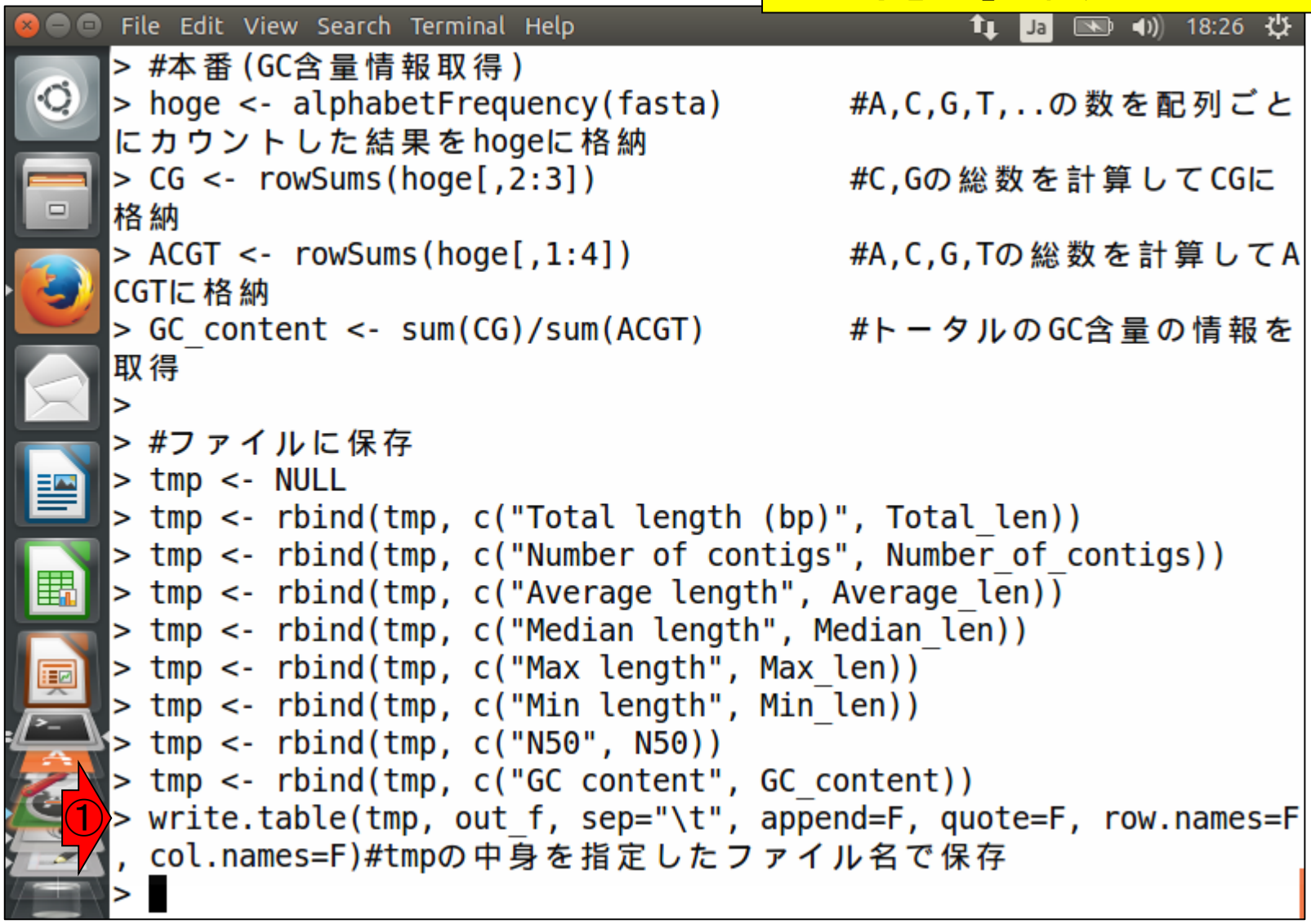
W9-5：コピペ

Rの画面上で①ペースト。ホスト－ゲスト間でコピペがうまくできないときは、②のfirefoxを起動してコードのコピーをすればよい。あるいは一旦Bio-Linuxを保存せずに終了して再起動とか



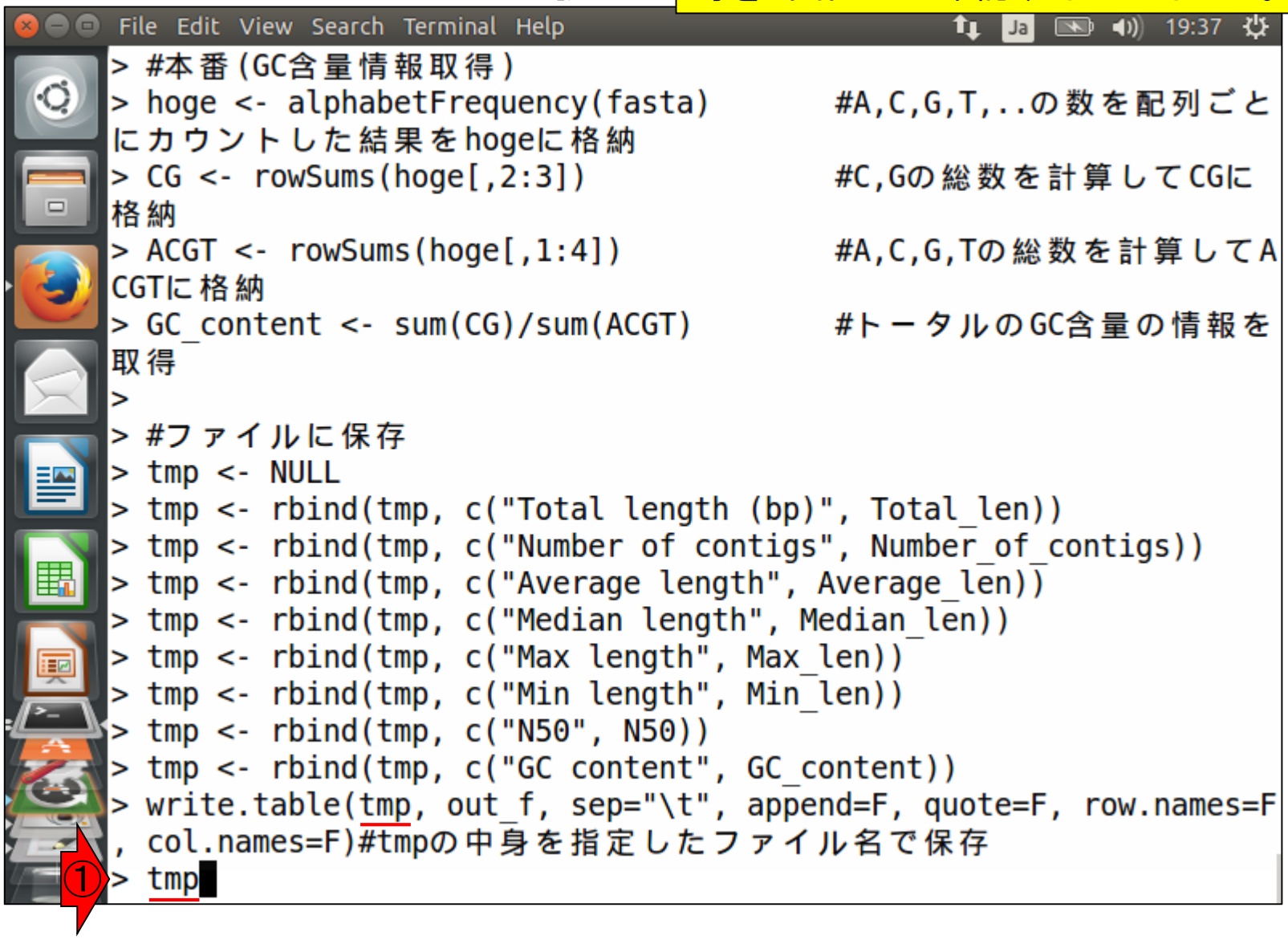
W9-6：コピペ後

エラーなく実行できたときの全貌（っていうか最後のほう）。①最後の行は、write.tableという関数を用いてtmpの中身をout_fで指定したファイルに保存するコマンド



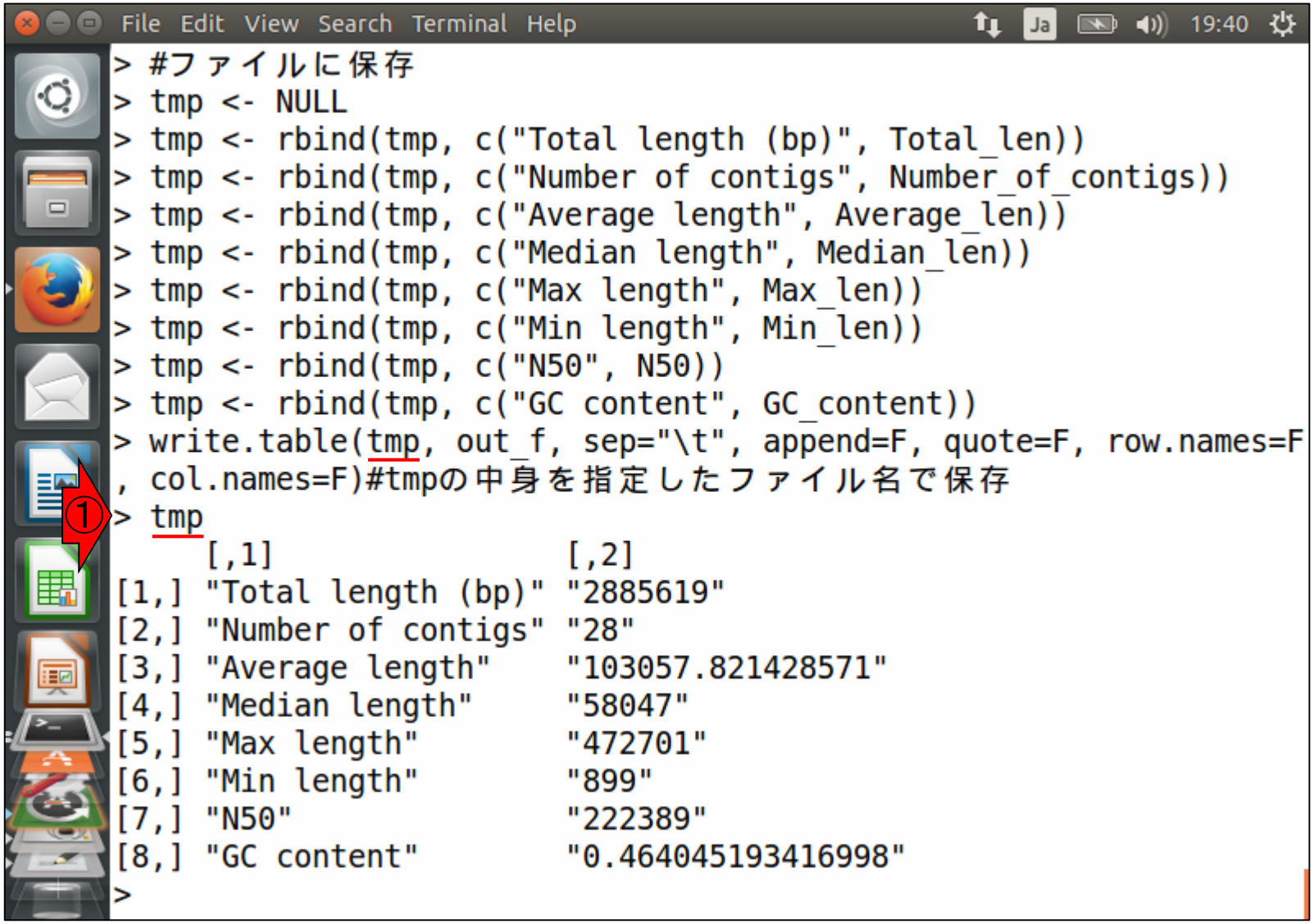
W9-6：コピペ後

それゆえ、out_fで指定した出力ファイル(result_JALAB1.txt)をイチイチ開いて確認しなくても、①tmpと打って、tmpの中身をR画面上で確認するのでもよい。ただの復習



W9-6:コピペ後

①tmpの中身を表示させた結果。これが「連載第1回の図2」と基本的に同じもの



①連載第1回の図2です

ほらね

```
> #ファイルに保存
> tmp <- NULL
> tmp <- rbind(tmp, c("Total le
> tmp <- rbind(tmp, c("Number o
> tmp <- rbind(tmp, c("Average
> tmp <- rbind(tmp, c("Median l
> tmp <- rbind(tmp, c("Max leng
> tmp <- rbind(tmp, c("Min leng
> tmp <- rbind(tmp, c("N50", N5
> tmp <- rbind(tmp, c("GC conte
> write.table(tmp, out_f, sep="
, col.names=F)#tmpの中身を指定し
> tmp
      [,1]                 [,2]
[1,] "Total length (bp)" "28856
[2,] "Number of contigs" "28"
[3,] "Average length"    "10305
[4,] "Median length"     "58047
[5,] "Max length"        "47270
[6,] "Min length"        "899"
[7,] "N50"               "22238
[8,] "GC content"        "0.464
>
```

R Console

```
> #ファイルに保存
> tmp <- NULL
> tmp <- rbind(tmp, c("Total length (bp)", Total_len))
> tmp <- rbind(tmp, c("Number of contigs", Number_of_contigs))
> tmp <- rbind(tmp, c("Average length", Average_len))
> tmp <- rbind(tmp, c("Median length", Median_len))
> tmp <- rbind(tmp, c("Max length", Max_len))
> tmp <- rbind(tmp, c("Min length", Min_len))
> tmp <- rbind(tmp, c("N50", N50))
> tmp <- rbind(tmp, c("GC content", GC_content))
> write.table(tmp, out_f, sep="\t", append=F, quote=F, row.names=F, col.names=$
> tmp
     [,1]                 [,2]
[1,] "Total length (bp)" "2885619"
[2,] "Number of contigs" "28"
[3,] "Average length"    "103057.821428571"
[4,] "Median length"     "58047"
[5,] "Max length"        "472701"
[6,] "Min length"        "899"
[7,] "N50"               "222389"
[8,] "GC content"        "0.464045193416998"
>
```

①

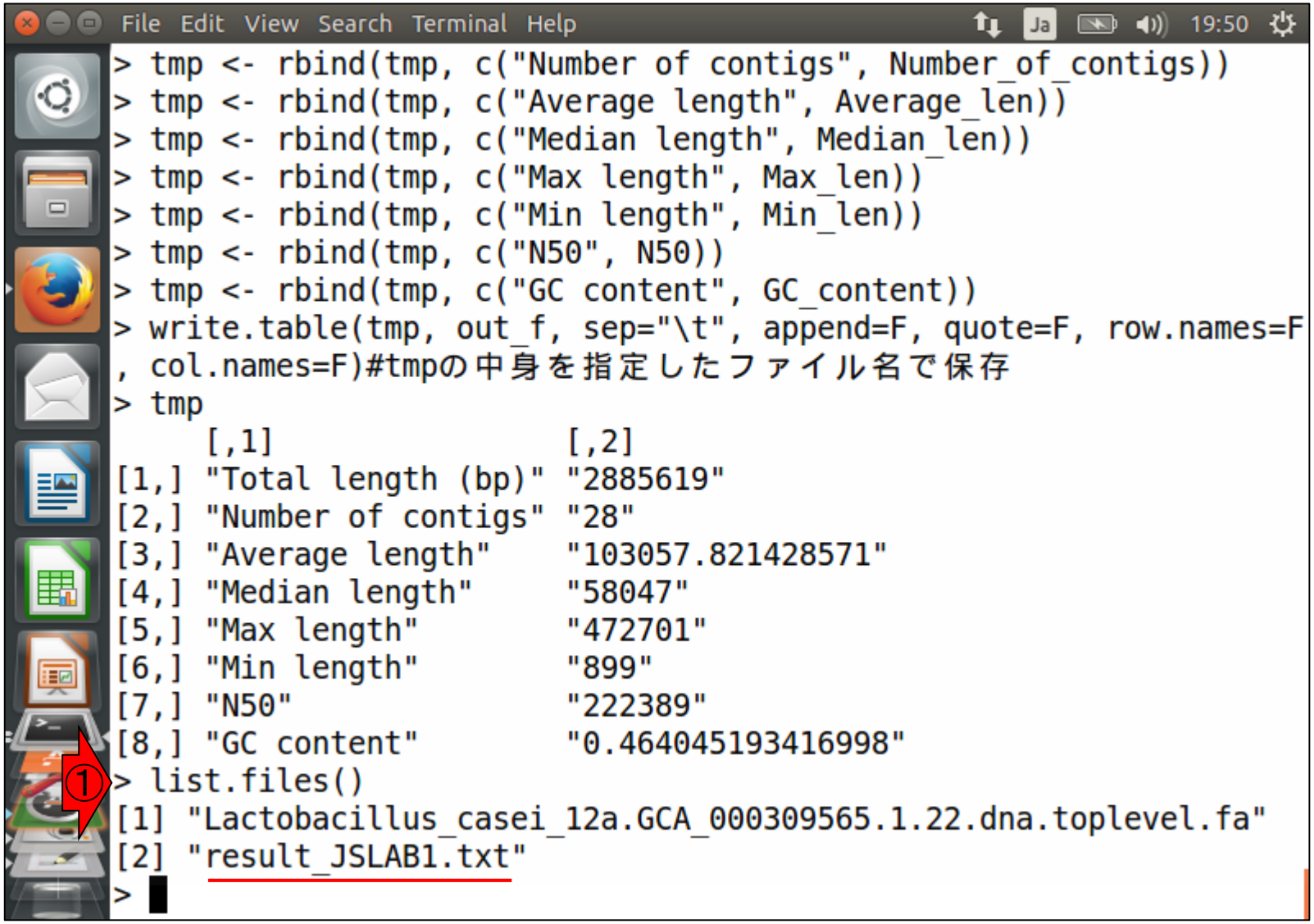
図 2. Rでゲノム配列解析。Rコード実行結果のスクリーンショットを示している。

解析するという手間が軽減されることが期
pression Atlas と似たウェブツールである
ている DBCLS の今後の活動に期待したい。

原著論文[36]の Table 1 の記載内容(コン
タル塩基数：2,885,619 bp、%GC：46.4
られていることがわかる。また、最長コ

W9-7：存在確認

①確かにW9-3のコピペ前には存在しなかったresult_JSLAB1.txtがある

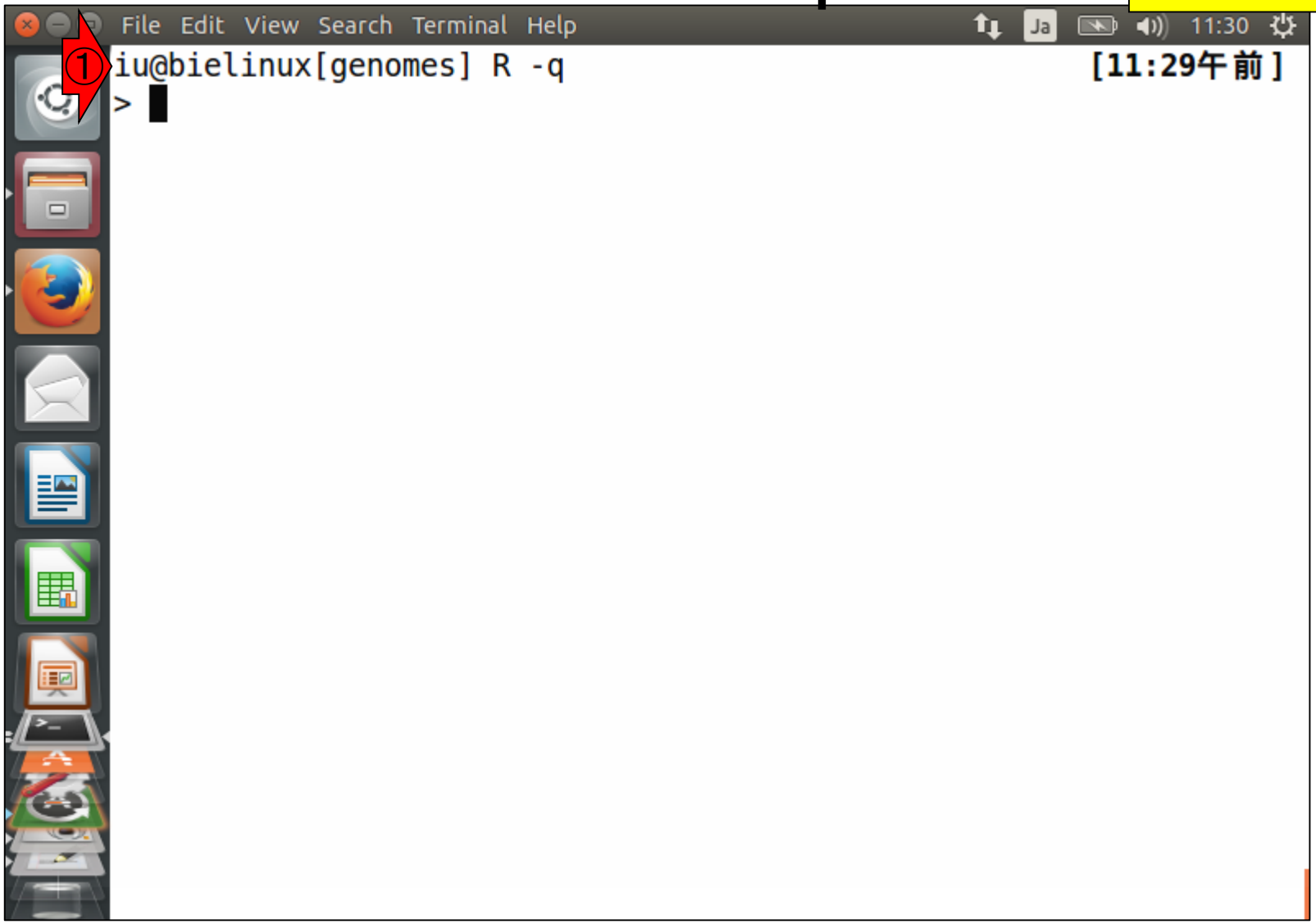


W9-7：存在確認

①Rを終了させて、②lsで
存在確認しているだけです

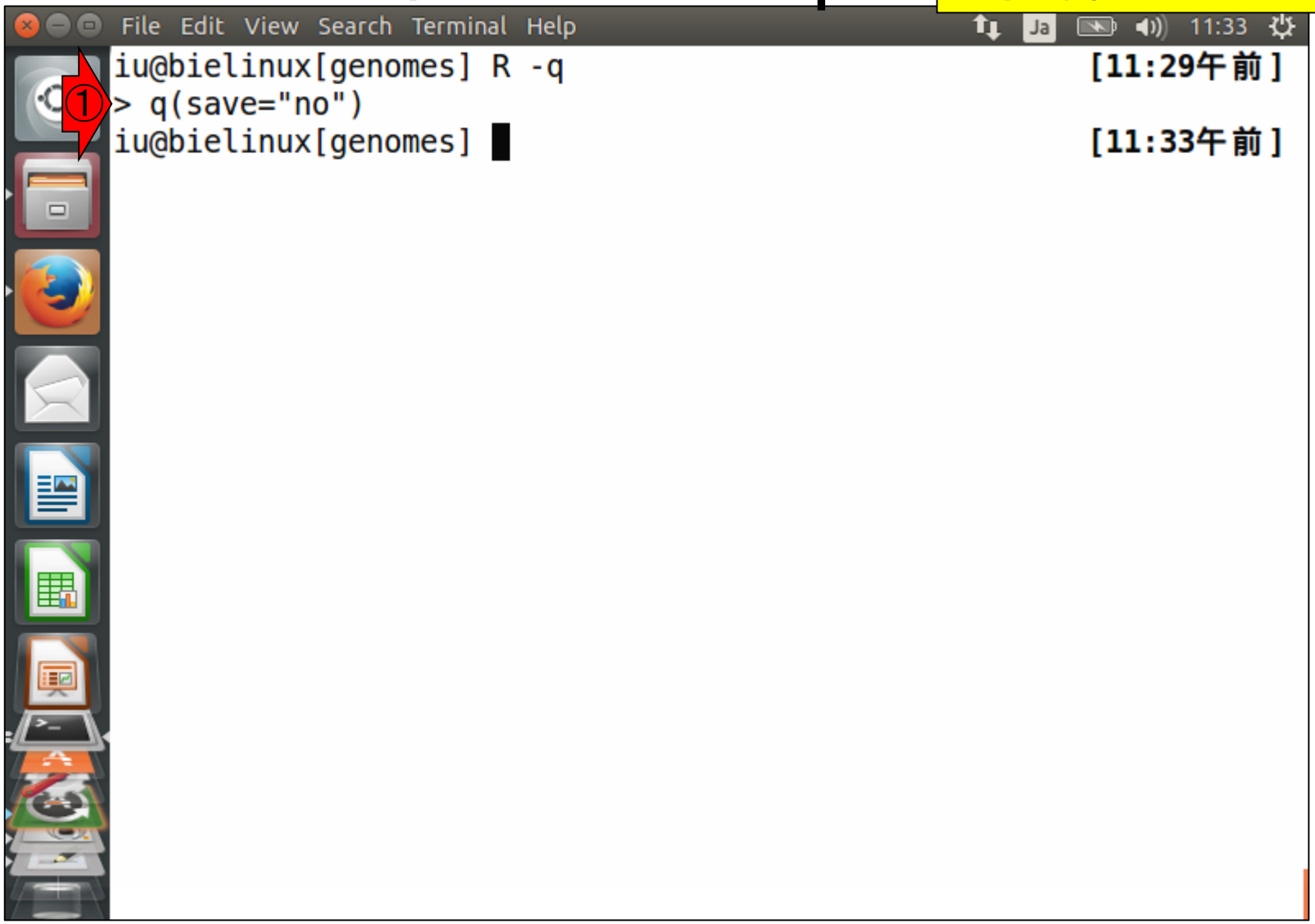
W9-8：R起動時のTips

①R起動時に-qオプションをつけることで、スタートアップメッセージを非表示にすることができる。



W9-8：R終了時のTips

①R終了時に「save="no"」オプションをつけることで、「Save workspace image? [y/n/c]」と毎回聞かれる苦行から解放される

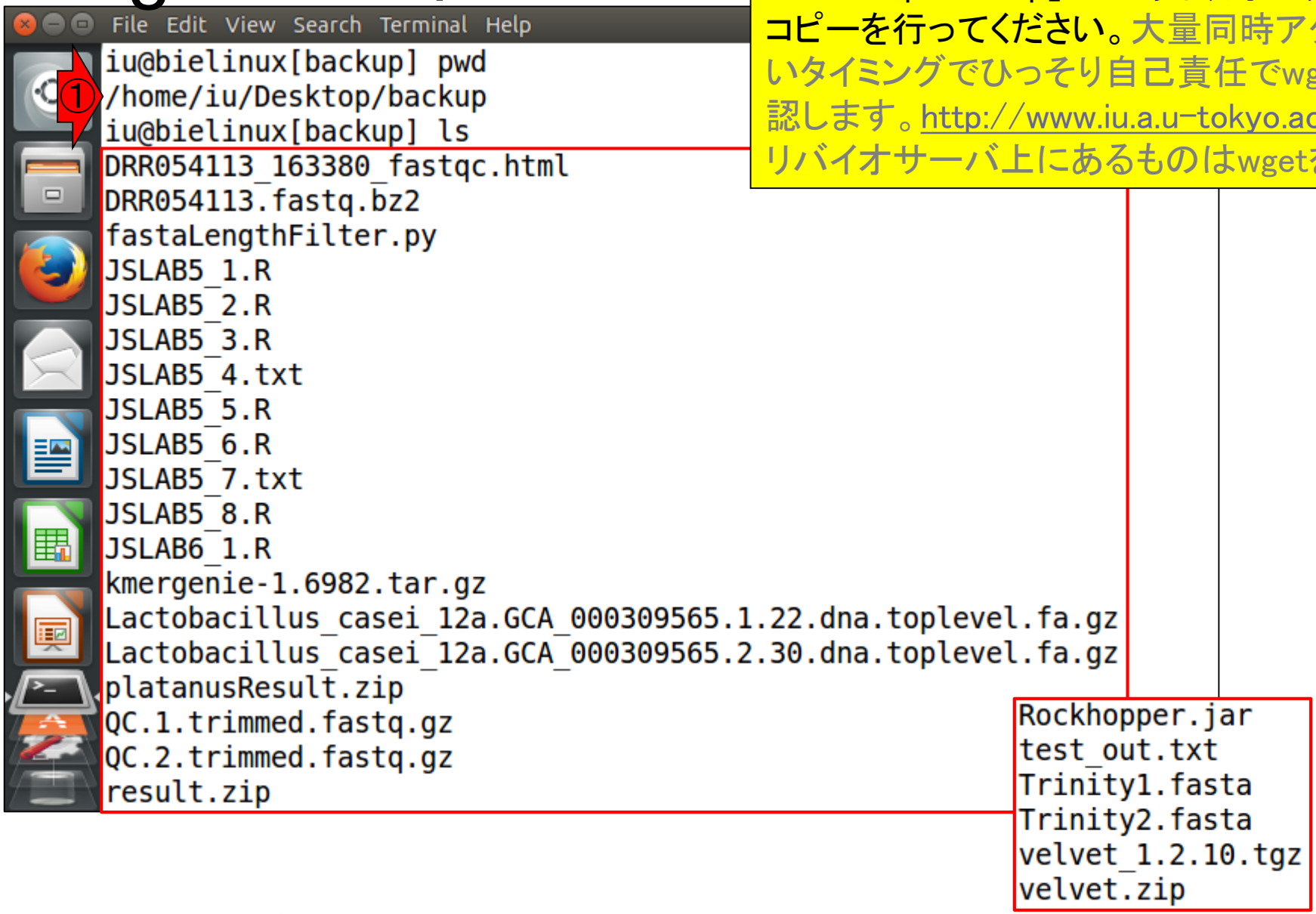


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

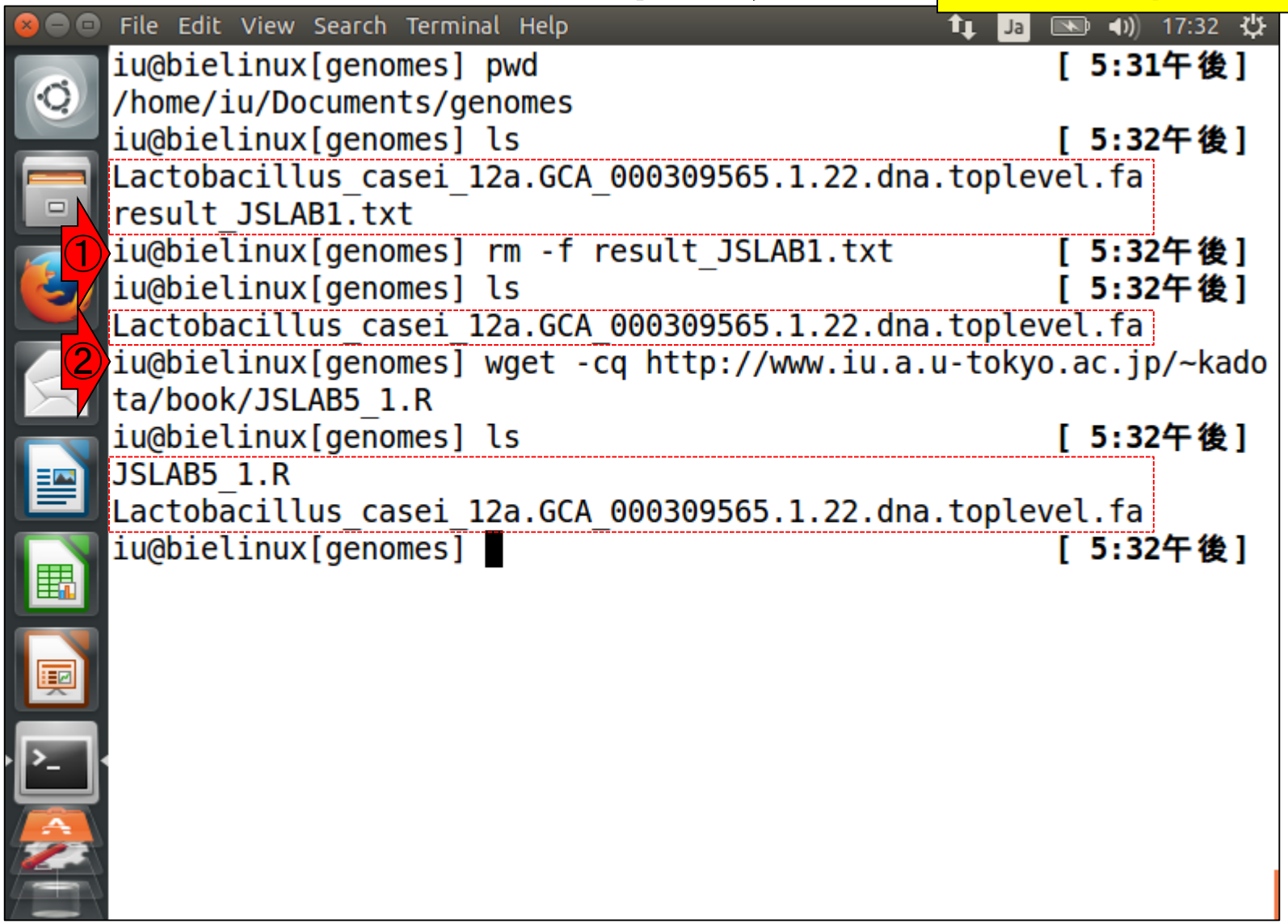
wgetは基本やらない

この後何回かwgetを利用しますが、基本的にはwgetしたつもりにしてください。実際に行うのは、①「~/Desktop/backup」上にある、対応するファイルのコピーを行ってください。大量同時アクセスにならないタイミングでひっそり自己責任でwgetするのは黙認します。http://www.iu.a.u-tokyo.ac.jp以下のアグリバイオサーバ上にあるものはwgetを試みてもOK



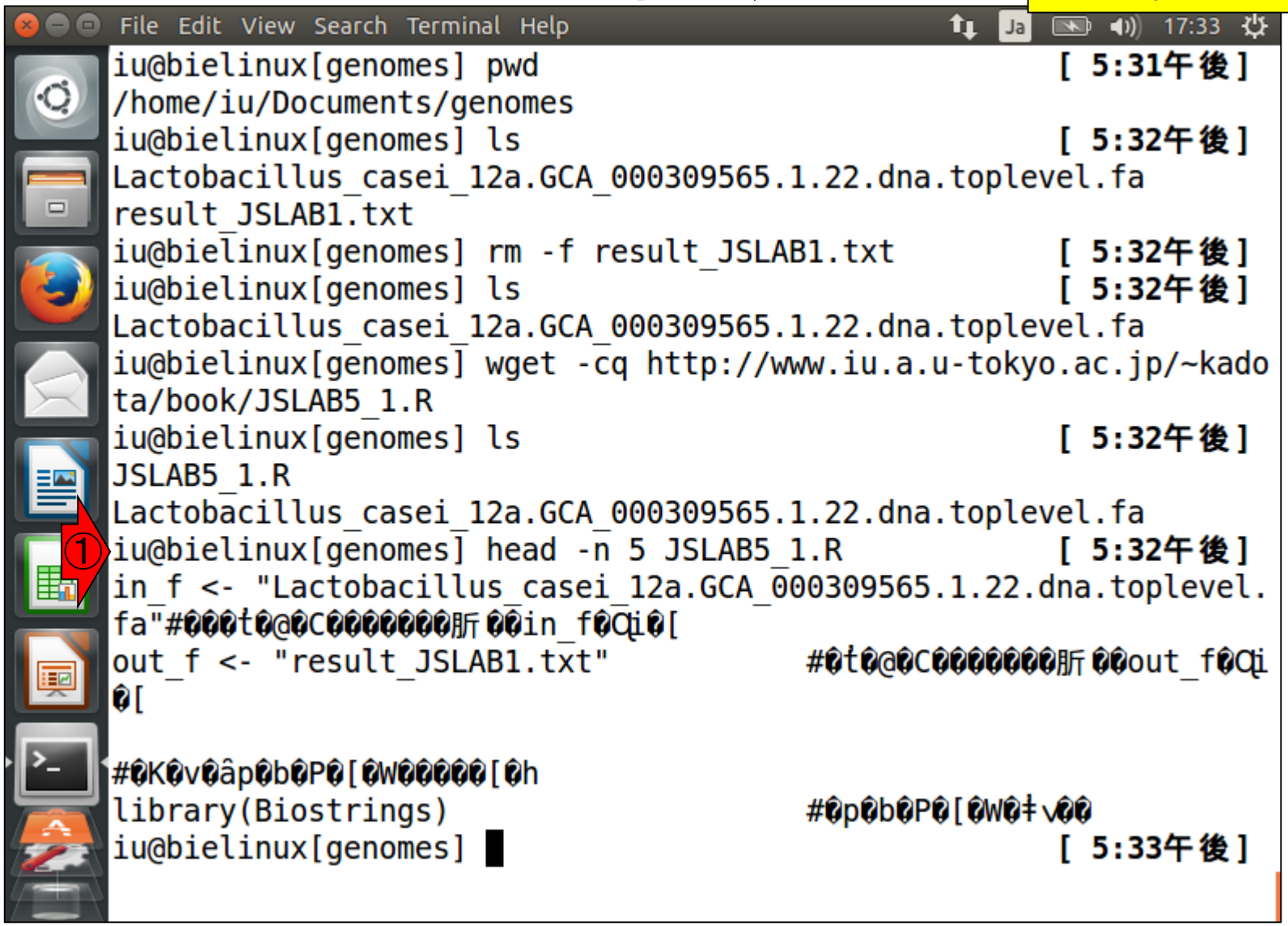
W10-1：source関数

①一旦result_JSLAB1.txtを削除。②wgetでJSLAB5_1.Rファイルを取得。JSLAB5_1.Rファイルの中身はW9-4（スライド131）と同じ



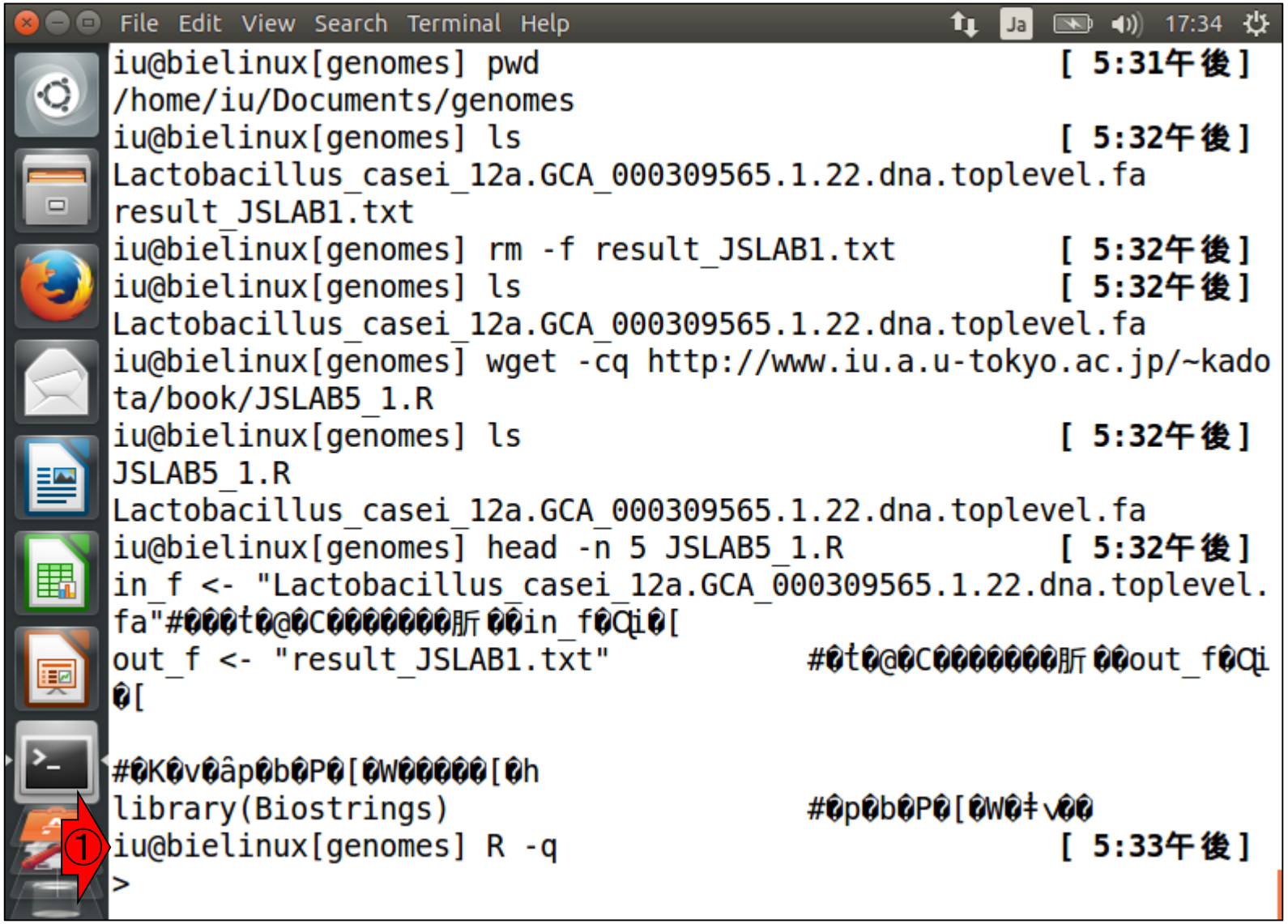
W10-1：source関数

①headで最初の5行分を表示。文字化けしているが、結果に影響しないコメント部分なので、ここでは気にしない



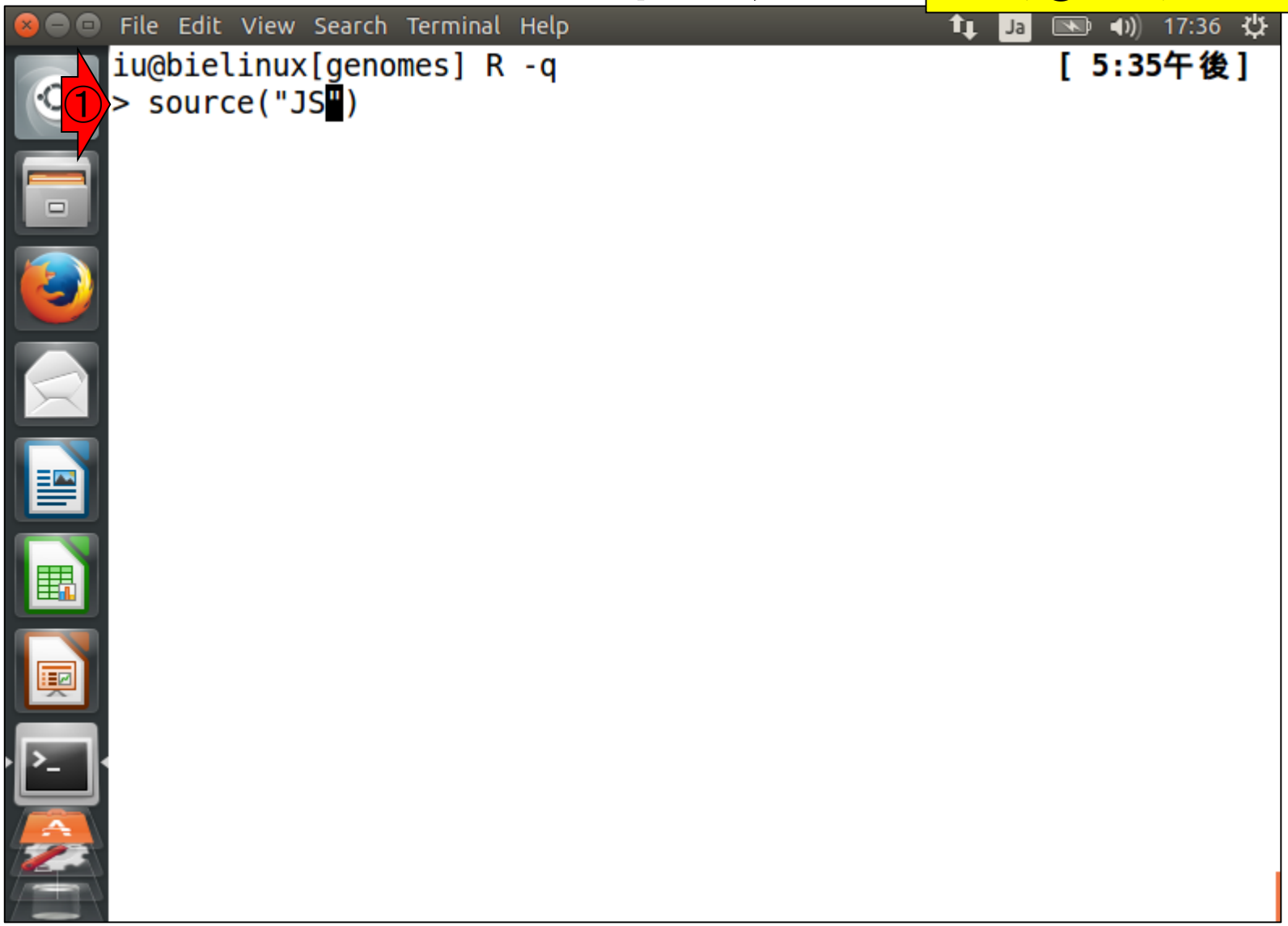
W10-1：source関数

①スタートアップメッセージを非表示にしてRを起動



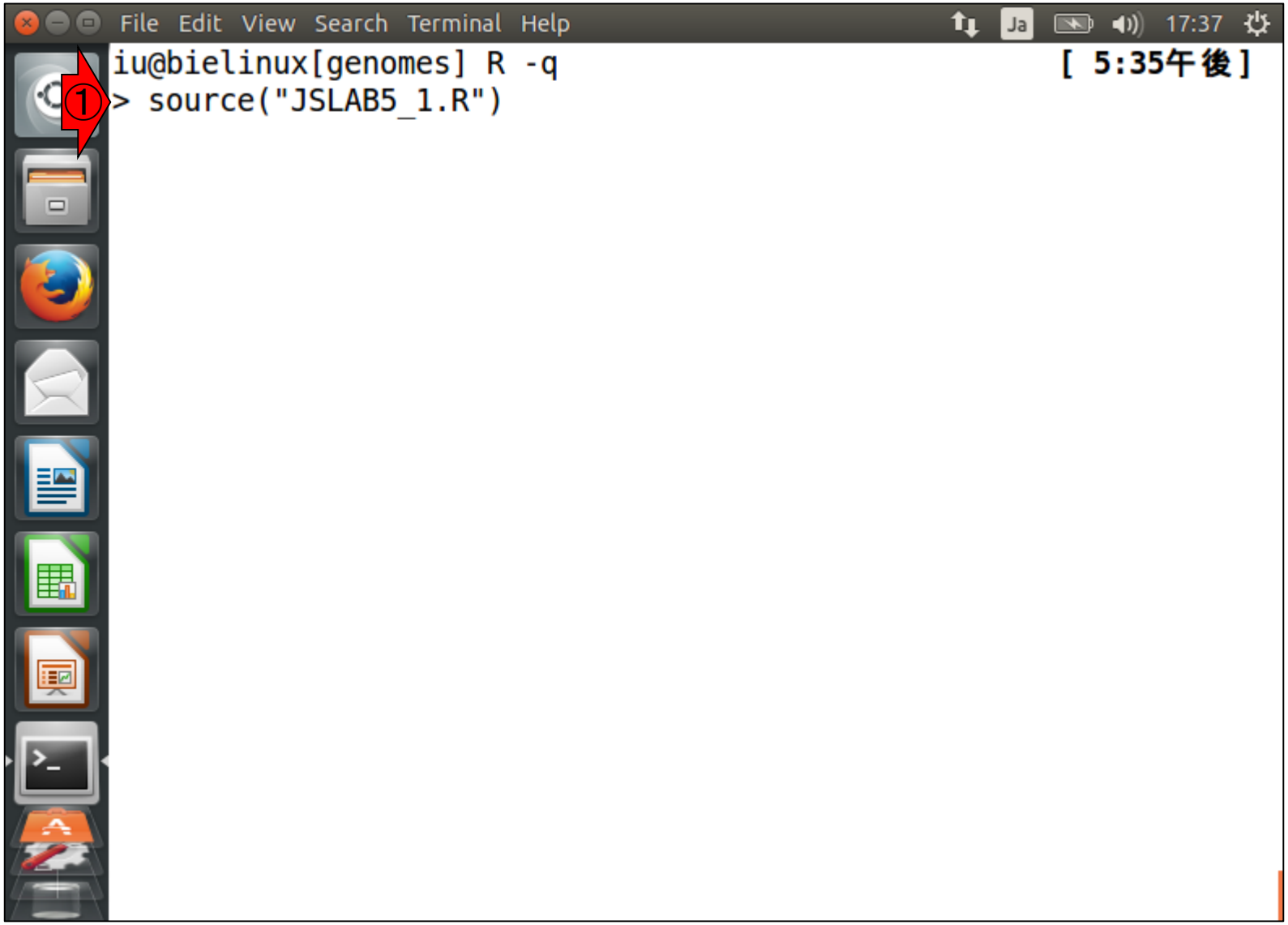
W10-1：source関数

目的は「source(“JSLAB5_1.R”)」と打ち込むこと。Rの対話モードでもタブ補完が有効なので、①のような状態でTabキーを押すと…



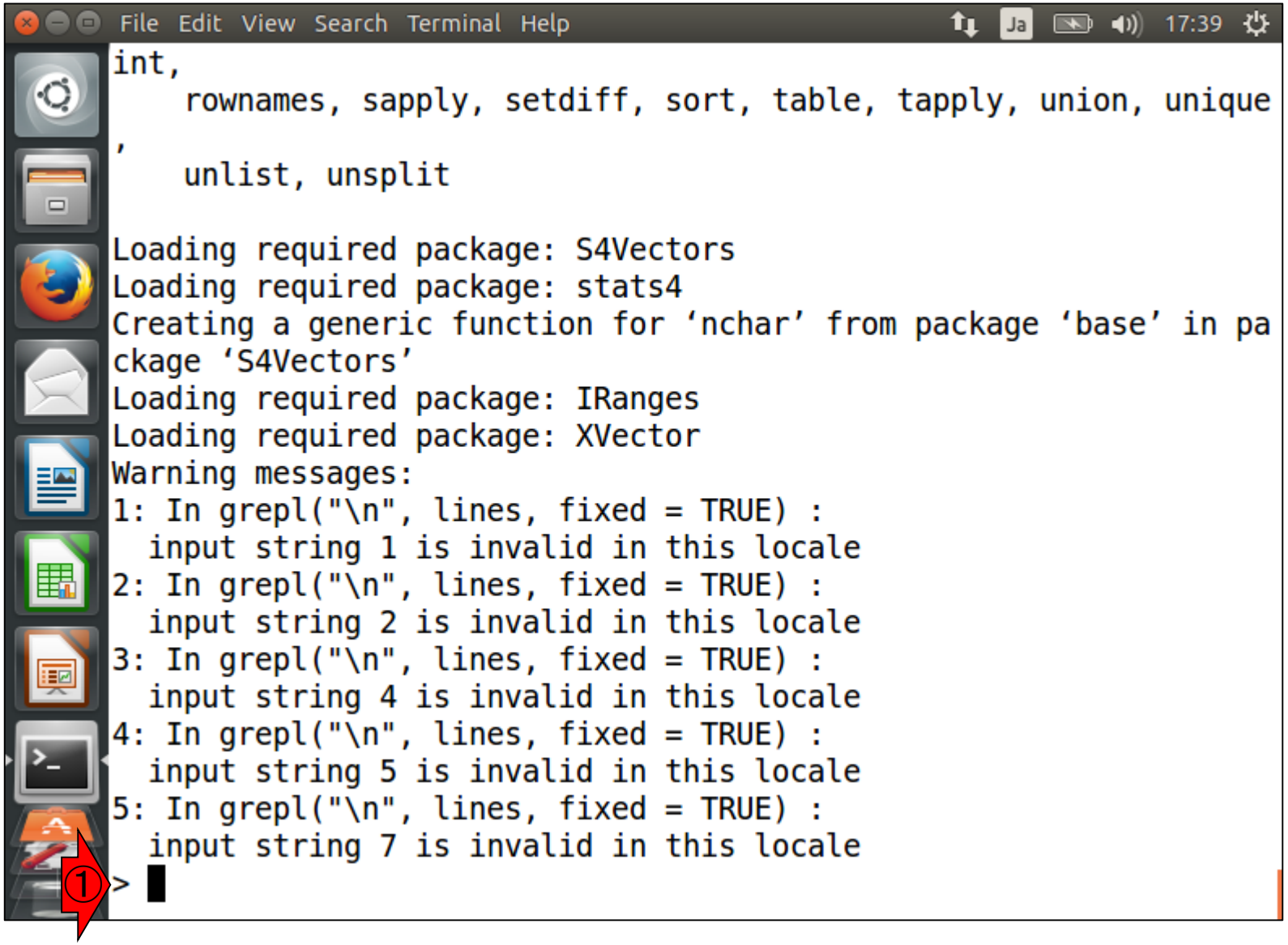
このようにタブ補完される。リターン

W10-1：source関数



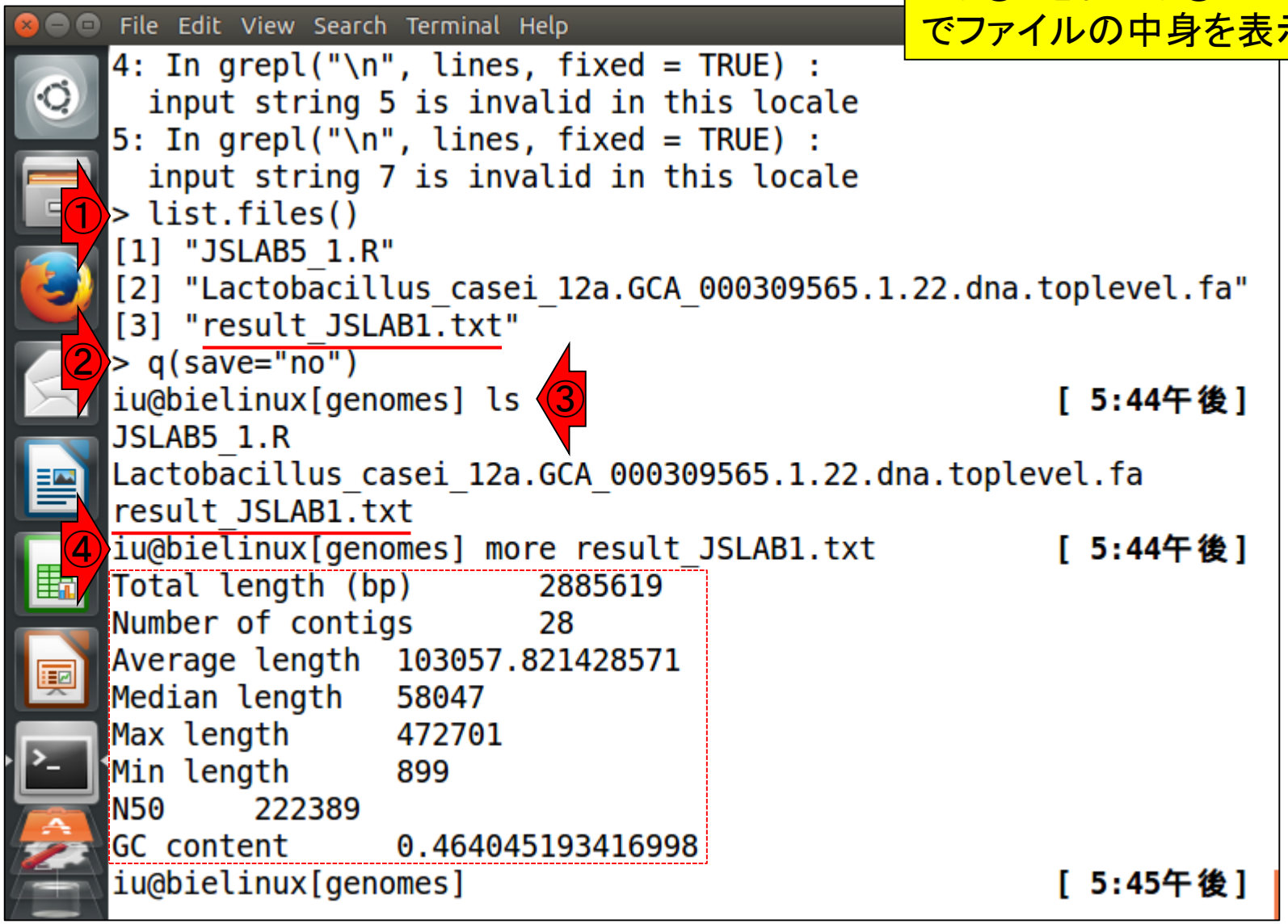
このような画面になる

W10-2：実行結果

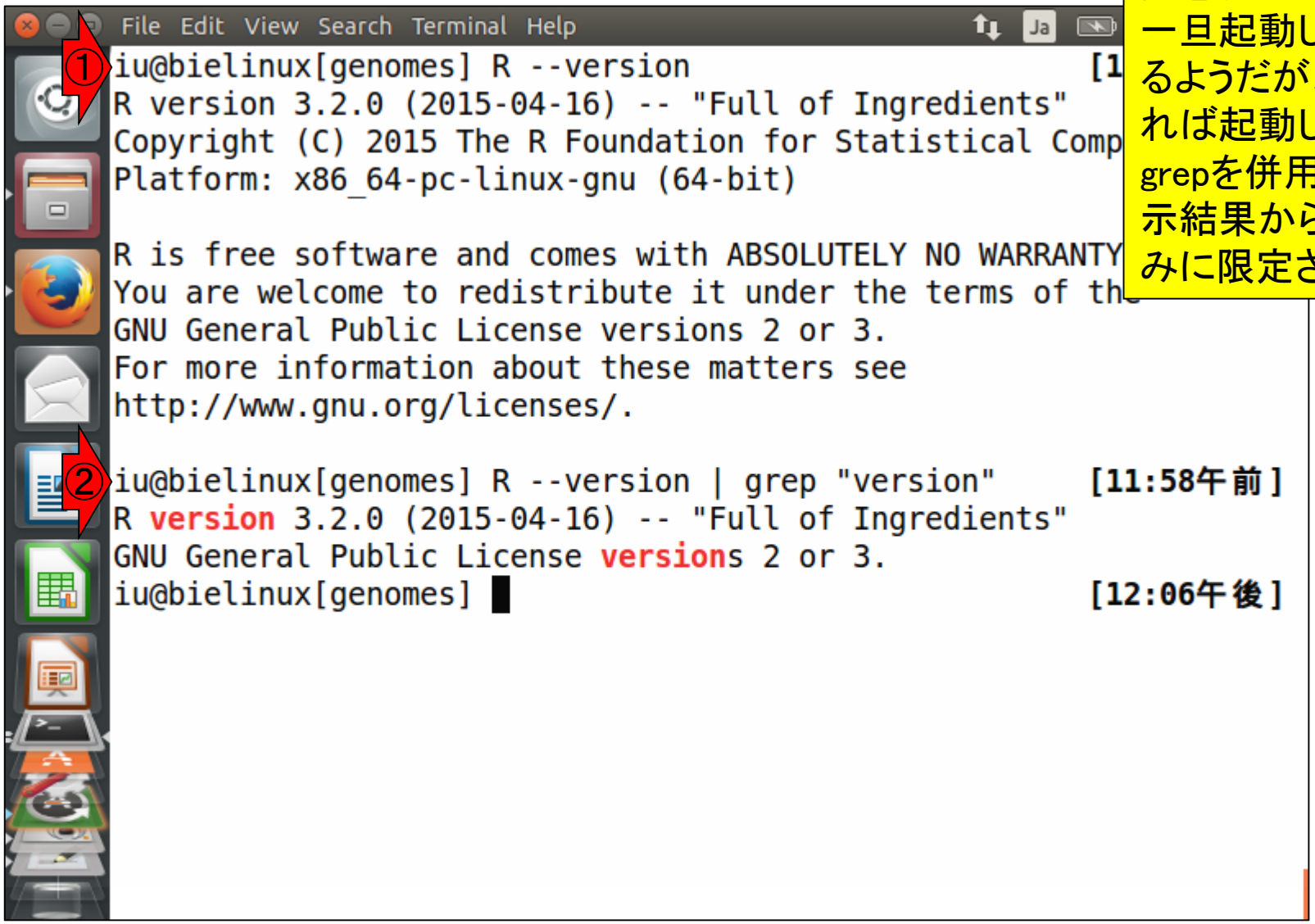


W10-3：確認

①list.files()で確認。確かに出力ファイルとして指定したresult_JSLAB1.txtが生成されている。②Rを終了。③lsで念のため確認。④moreでファイルの中身を表示。妥当な結果である



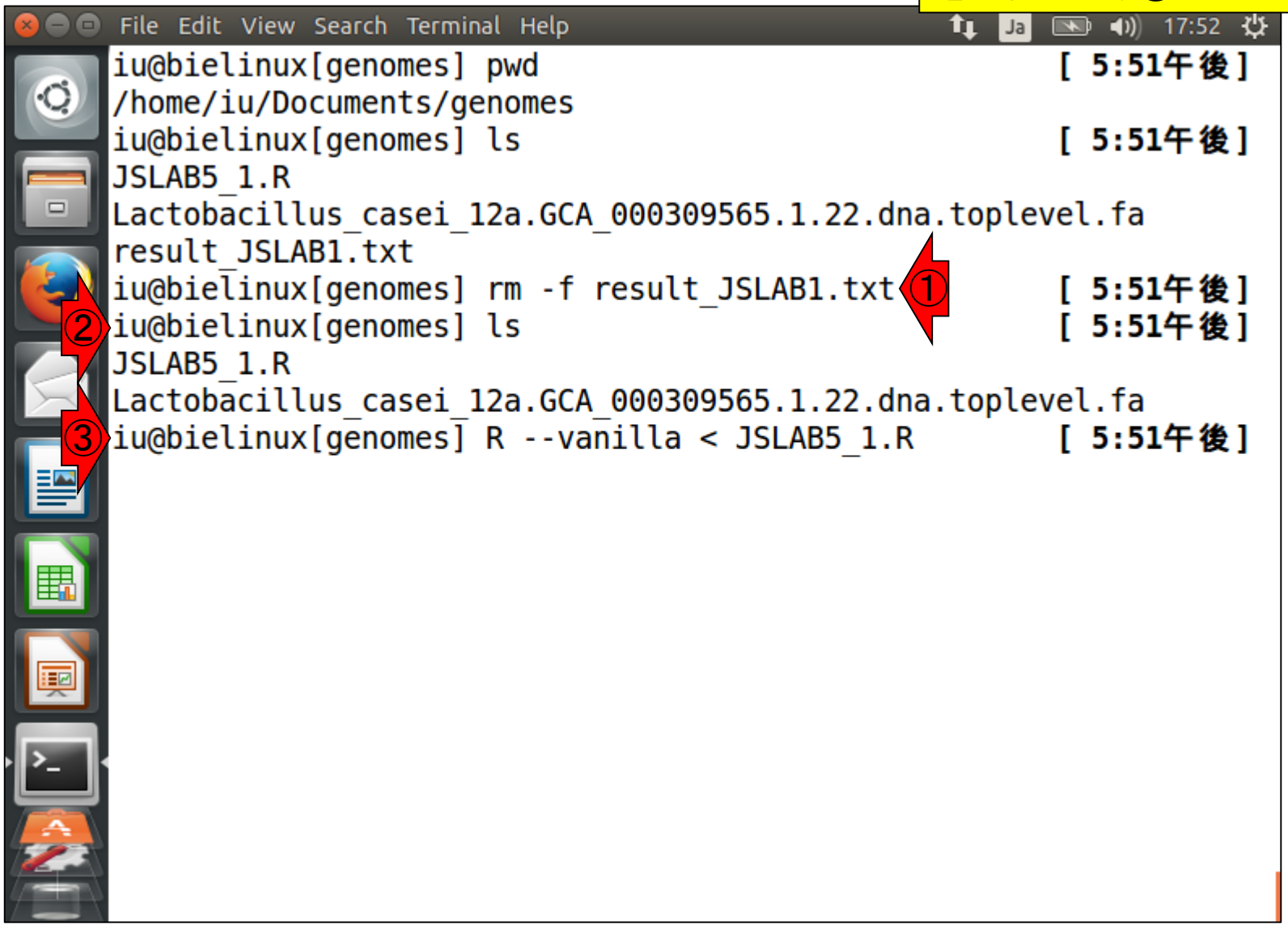
W10-4：Rのバージョン確認



①「R --version」と打つことで、Rを起動することなくバージョン確認をすることができる。正確には一旦起動してすぐに終了しているようだが、エンドユーザからすれば起動していないのと同じ。②grepを併用することで、さらに表示結果から”version”を含む行のみに限定させることができる

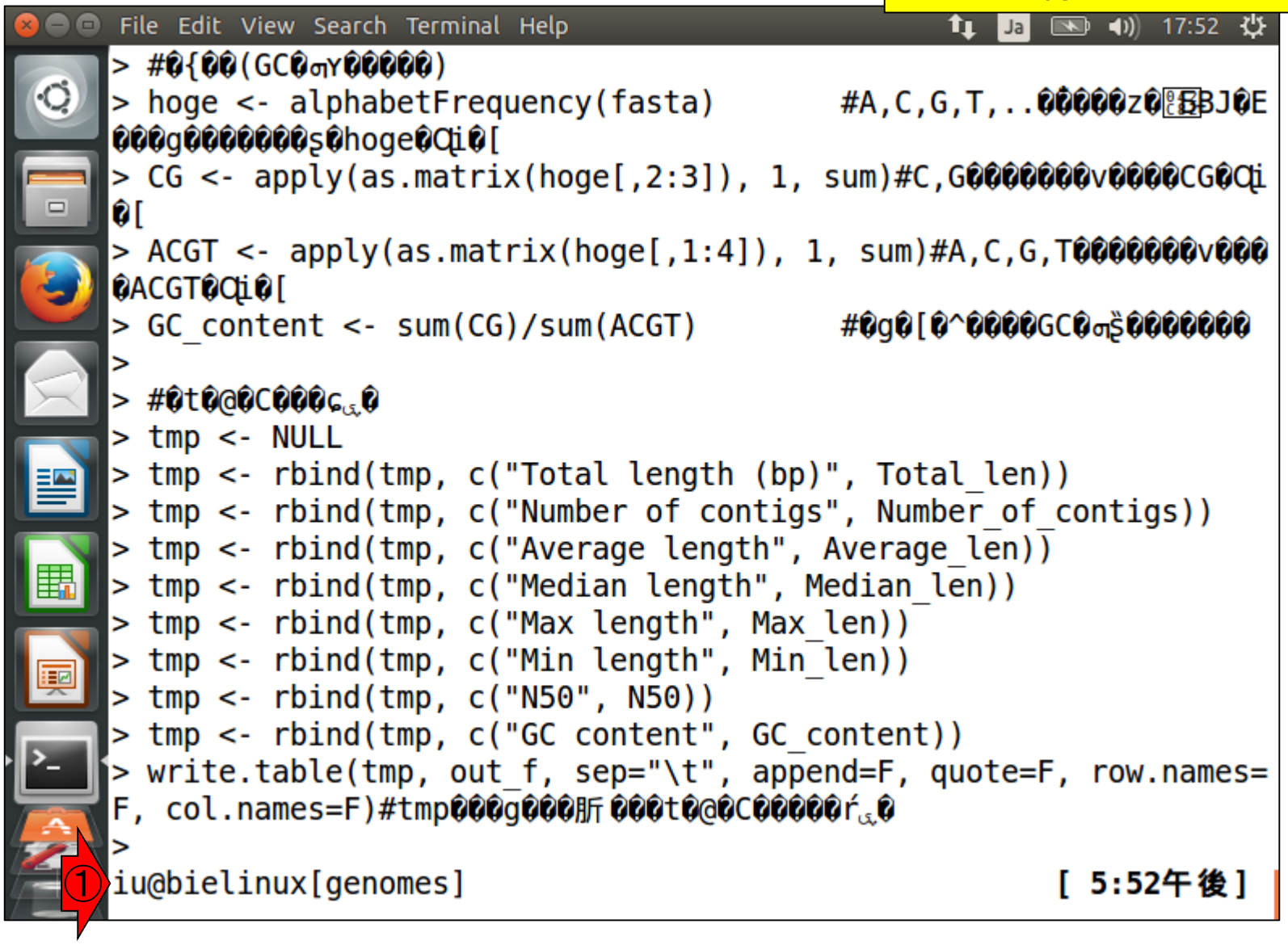
W10-5：バッチモード

①一旦result_JSLAB1.txtを削除。②lsで出力予定ファイル(result_JSLAB1.txt)がないことを確認して、③バッチモードの基本形を実行



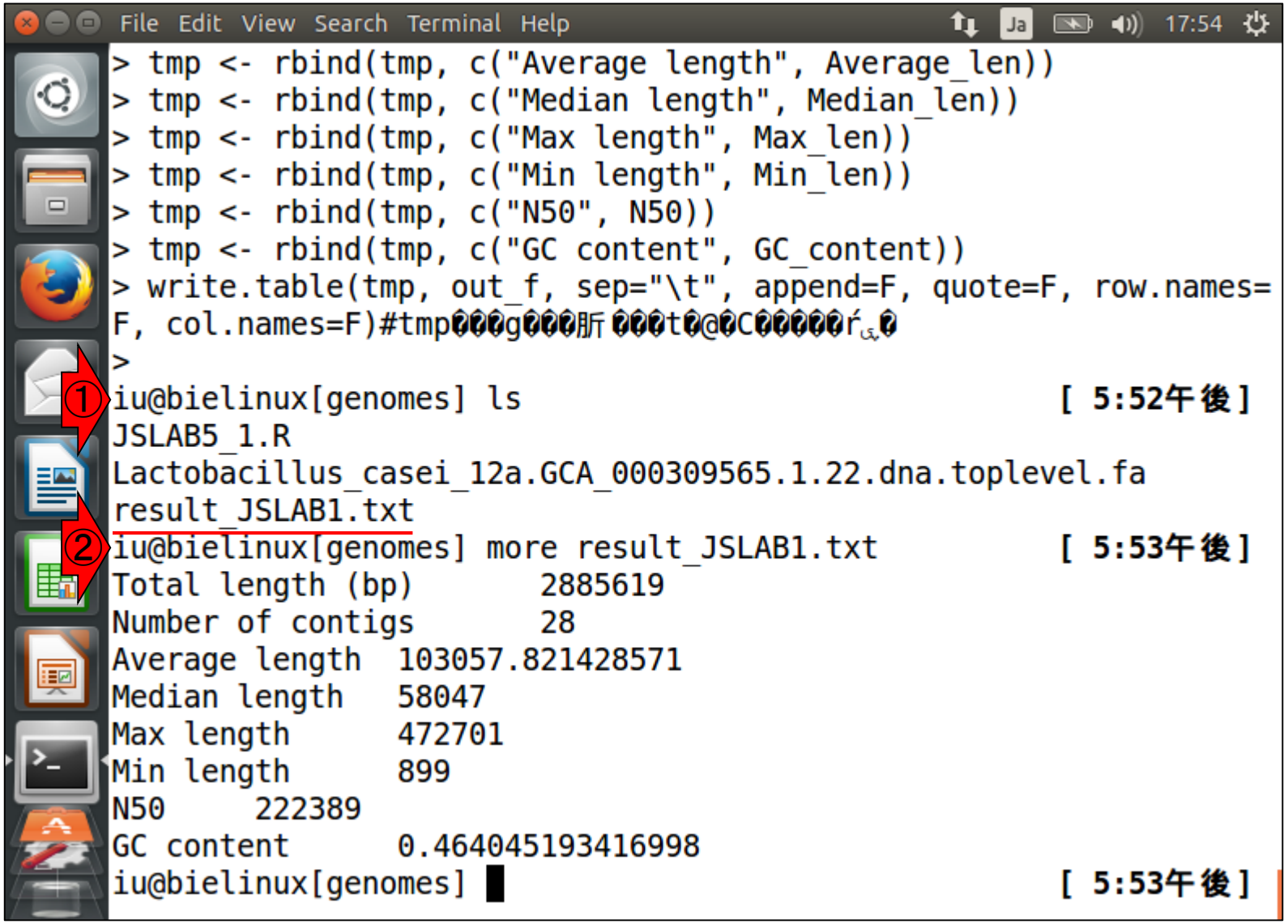
W10-5：バッチモード

①リターンを押して実行した結果。確かにq()などでRを終了させる手間なく、通常のLinuxコマンド入力待ち状態になっていることがわかる

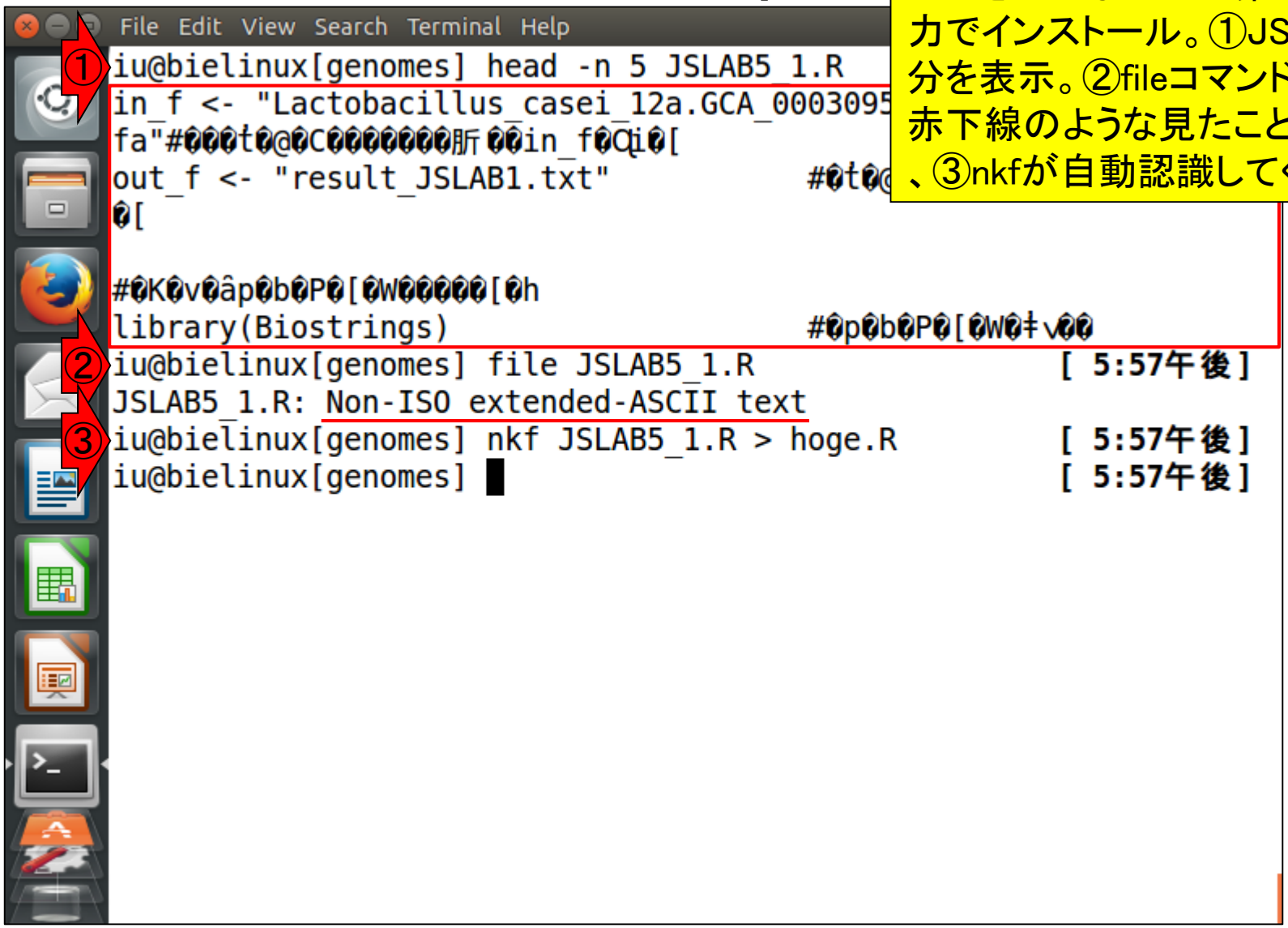


W10-5：確認

①lsで確認。確かにresult_JSLAB1.txtが出力結果として得られており、②その中身も妥当



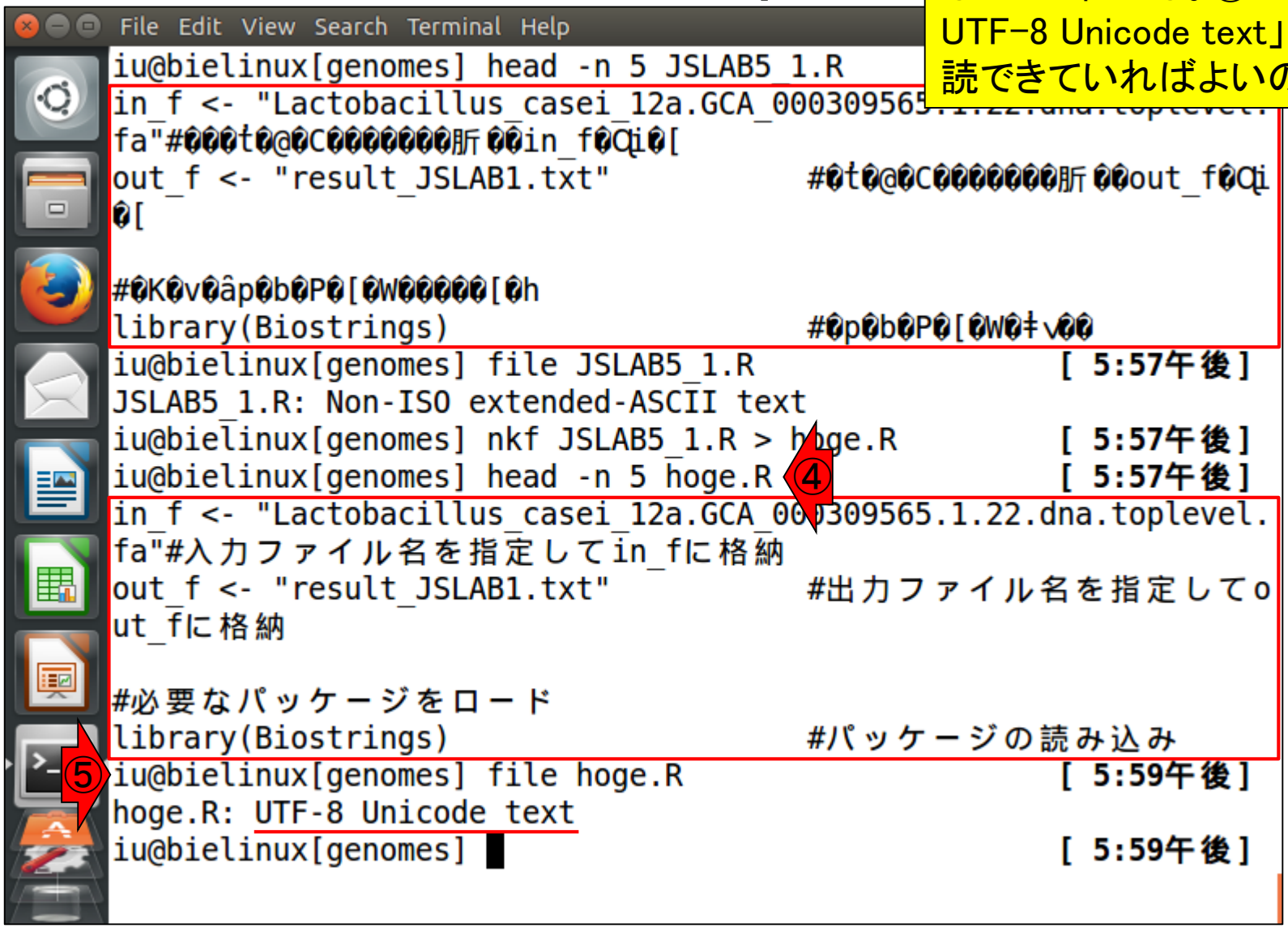
W11：文字化け対策



気になるヒト用に文字化け対策の復習。連載第4回W13-6にも記載あり。nkfのインストールができていない人は第4回W13-5を参照して自力でインストール。①JSLAB5_1.Rの最初の5行分を表示。②fileコマンドで文字コードを確認。赤下線のような見たこともない形式のようだが、③nkfが自動認識してくれるので気にもしない

W11：文字化け対策

④nkf実行後のファイルhoge.Rの最初の5行分を表示。確かに文字化けが解消されていることがわかる。⑤hoge.Rの文字コードは「UTF-8 Unicode text」というものらしいが、判読できていればよいので、特に気にしない

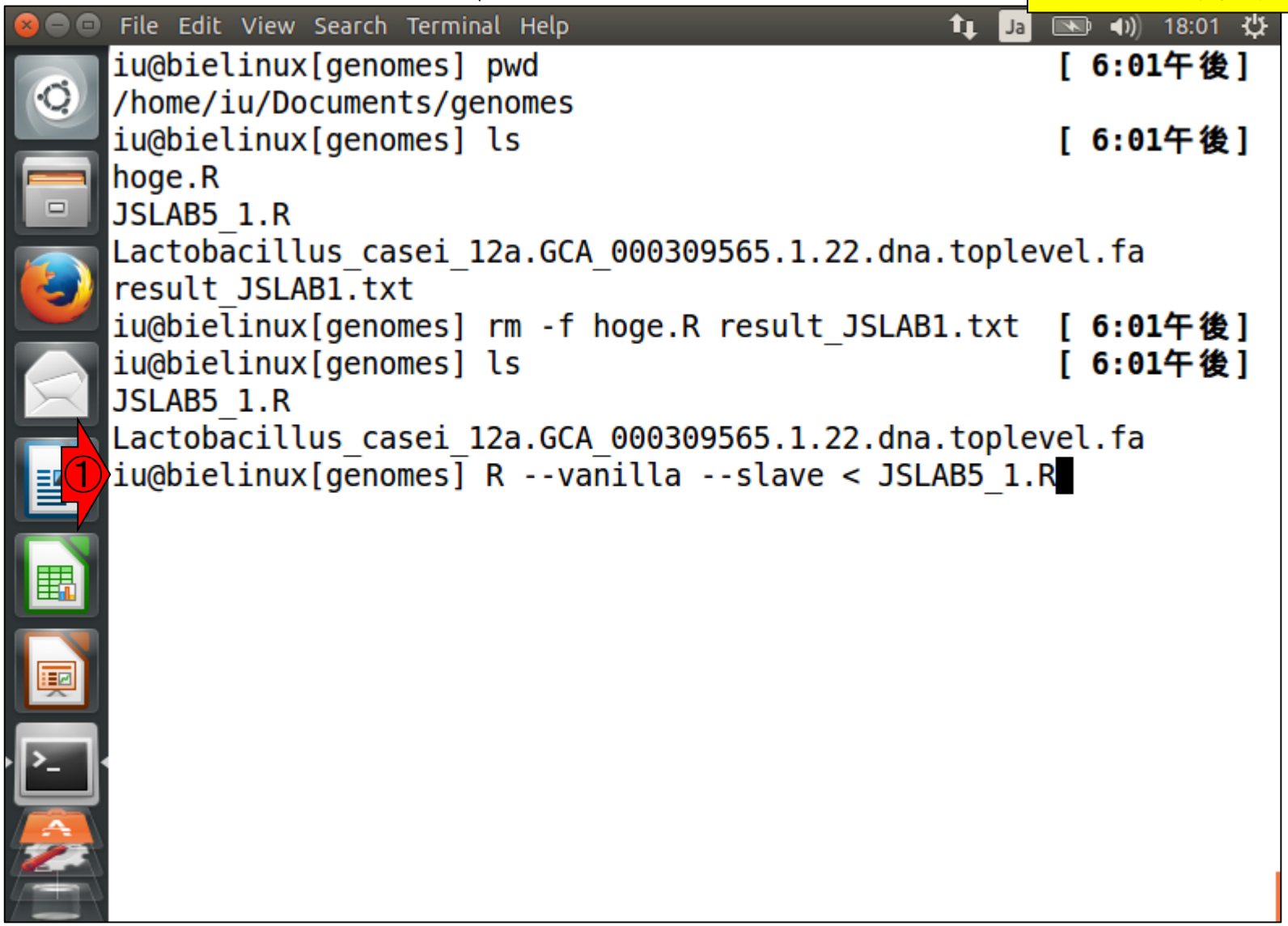


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - **W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定**
  - W13：gzip圧縮状態での利用

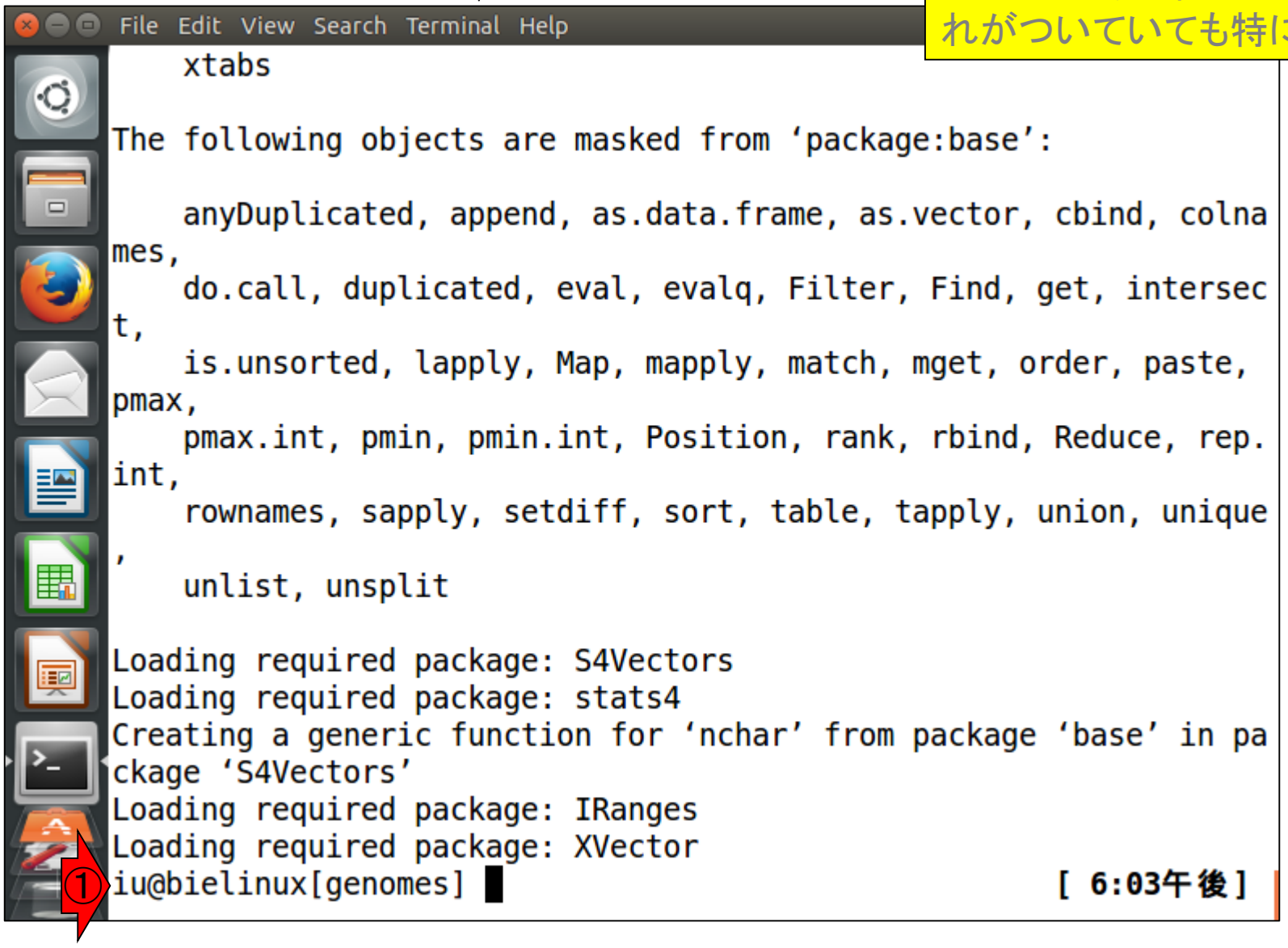
W12-1：発展形1

①--slaveオプションをつけて実行させると、実行中に画面表示されるものが減るので、多少見づらさが緩和される



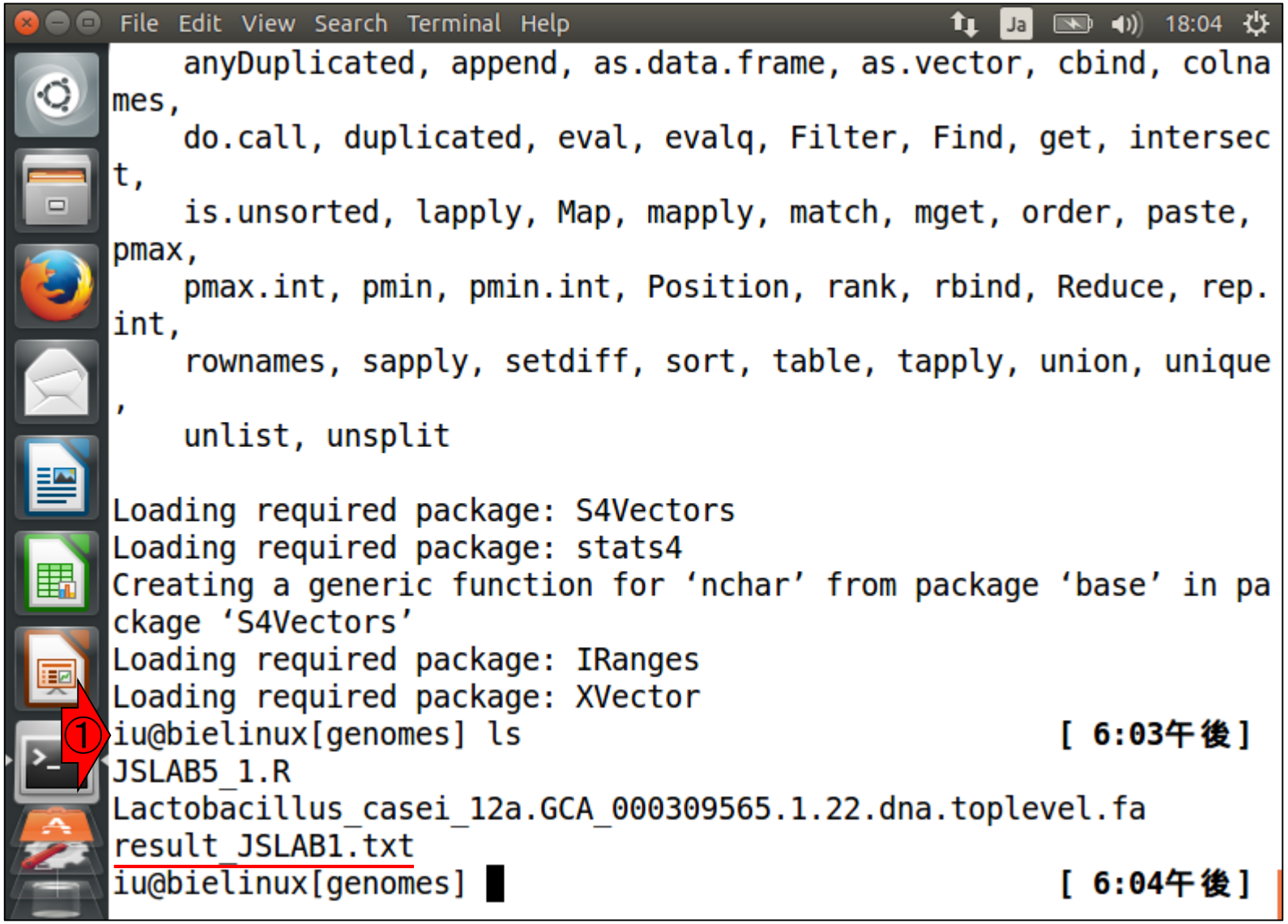
W12-1：発展形1

実行結果。W10-5と比較すると違いがわかる。違いはよくわからないかも…。--slaveオプションは、お約束的につけるものらしい。これがついていても特に気にする必要はない



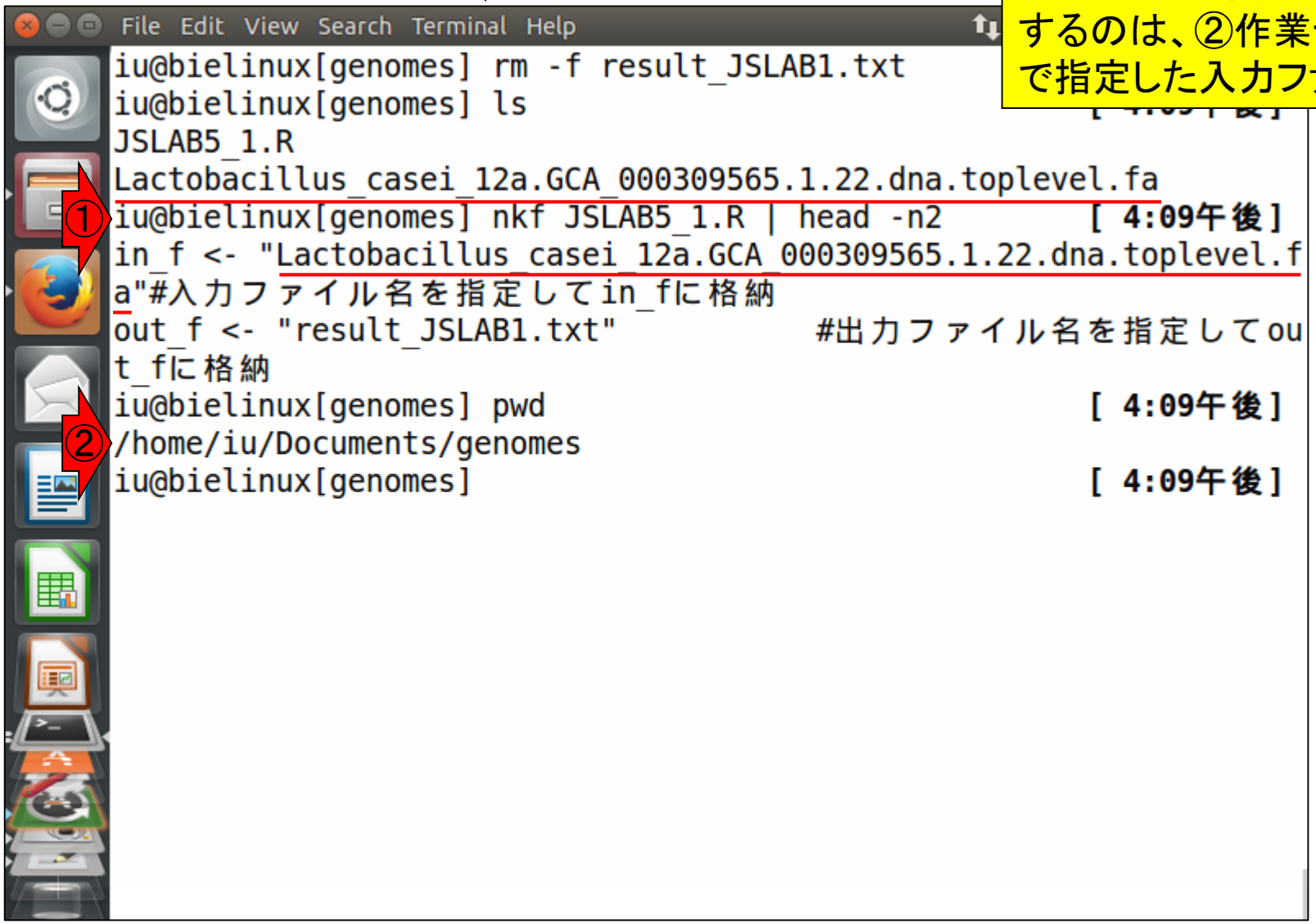
①lsで確認。確かに出力ファイルが存在する

W12-1：発展形1



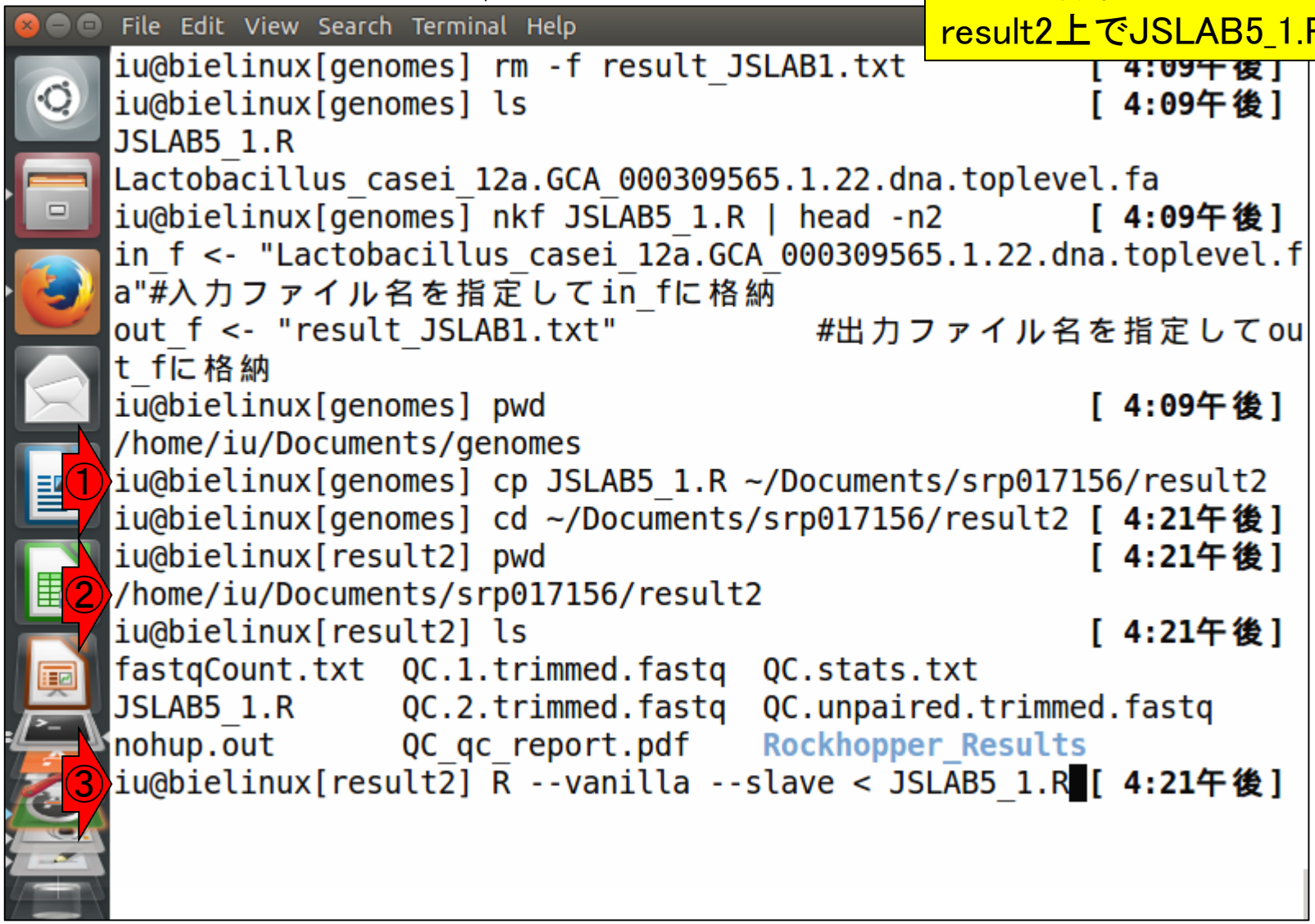
W12-2：発展形2

①JSLAB5_1.Rの最初の2行分を表示。nkf実行結果とパイプさせているのは、文字化け対策。JSLAB5_1.Rが正常動作するのは、②作業ディレクトリ上に、in_fで指定した入力ファイルが存在するから



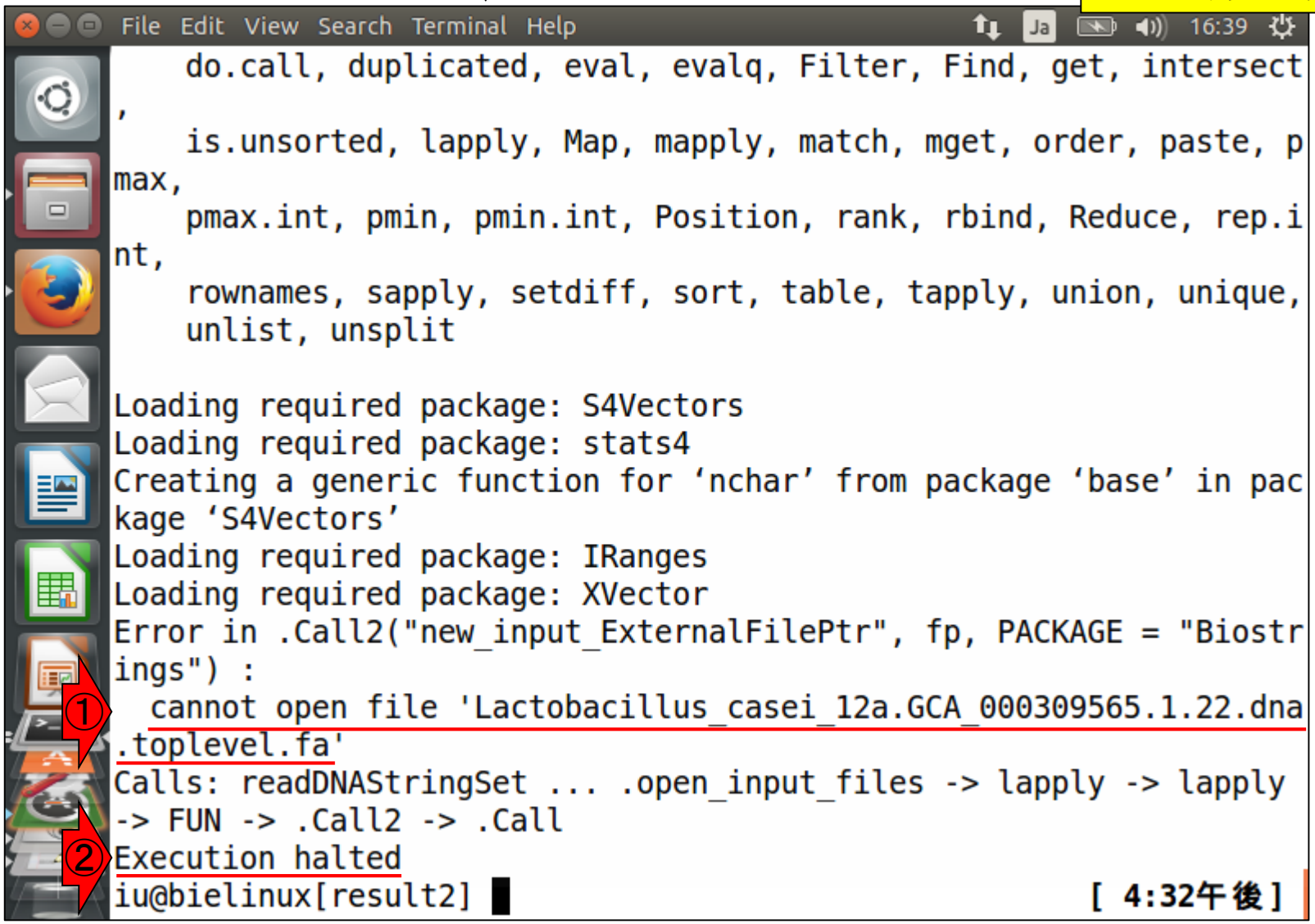
W12-2：発展形2

①result2ディレクトリにJSLAB5_1.Rをコピー。
②移動先にはJSLAB5_1.Rが入力として読み込む乳酸菌ゲノムファイルは存在しない。③result2上でJSLAB5_1.Rを実行してみると…



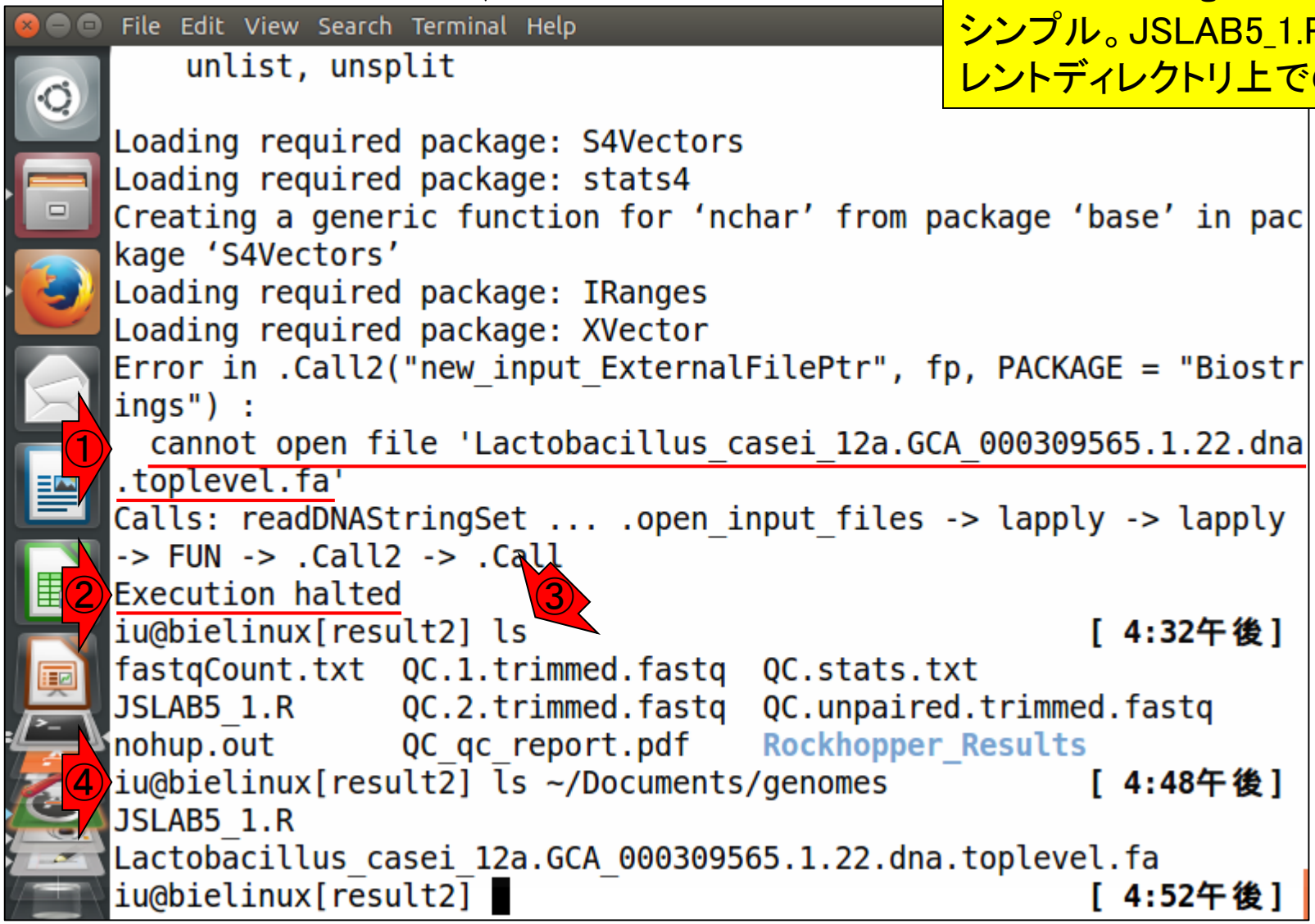
W12-2：発展形2

①cannot open file…や②Execution haltedというネガティブなメッセージからも、実行失敗の想像がつく



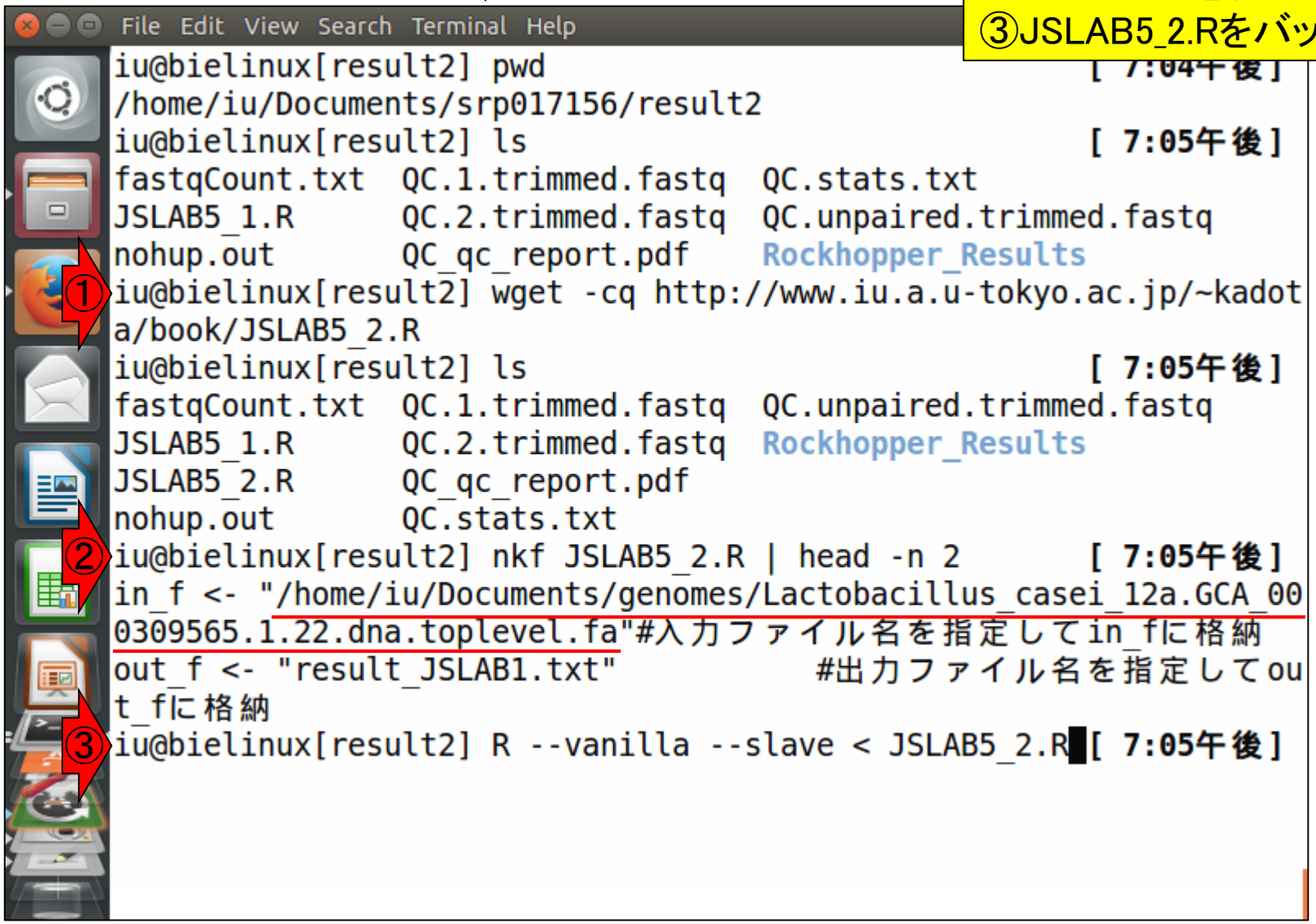
W12-2：発展形2

③lsした結果。result_JSLAB1.txtが生成されていないことがわかる。④当然~/Documents/genomes上にもない。理由はシンプル。JSLAB5_1.Rは、入力ファイルをカレントディレクトリ上でのみ探索しているから



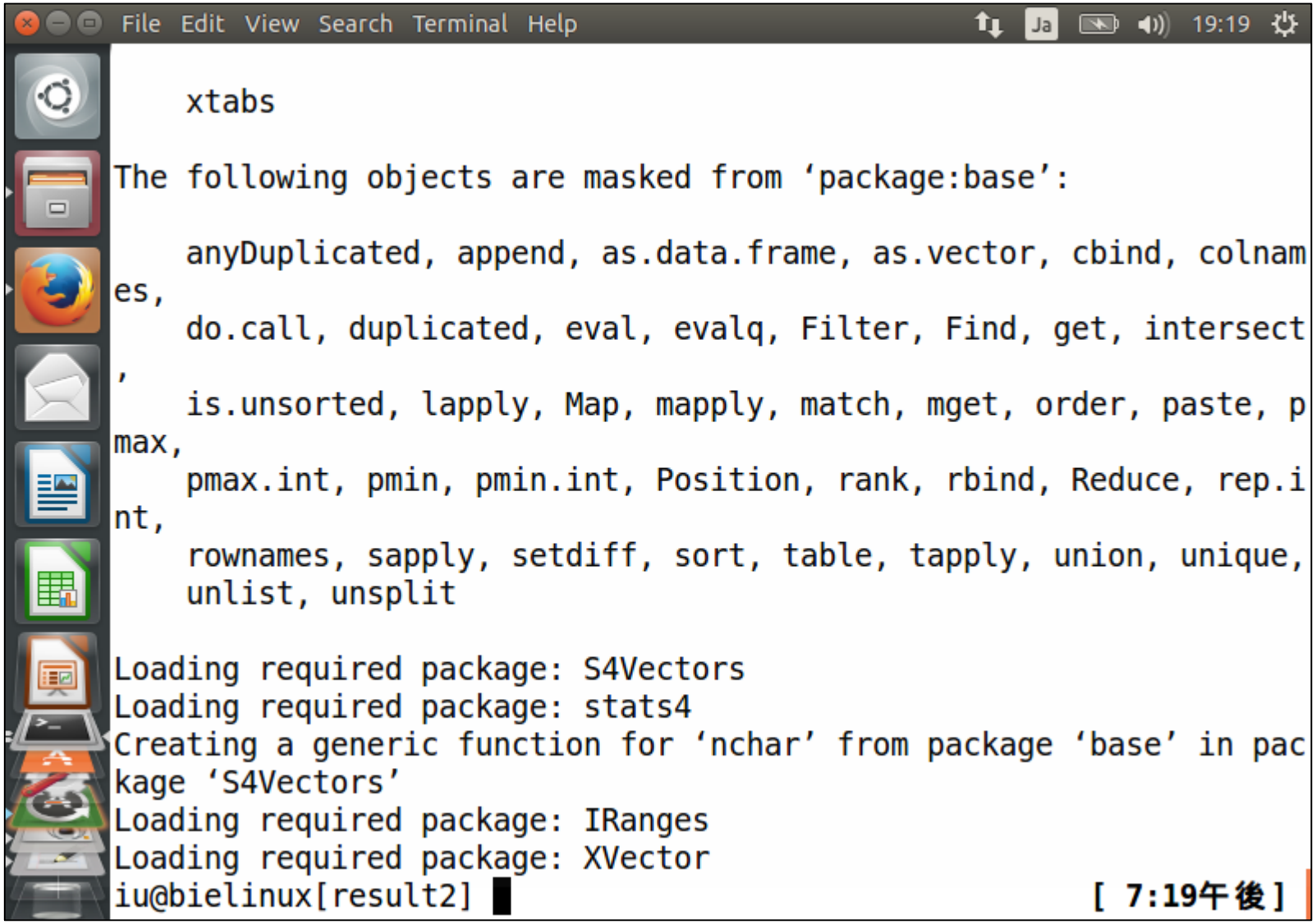
W12-3：発展形3

①wgetでJSLAB5_2.Rをダウンロード。②最初の2行分を表示。赤下線で示すように、入力ファイルを絶対パスで指定している。③JSLAB5_2.Rをバッチモードで実行



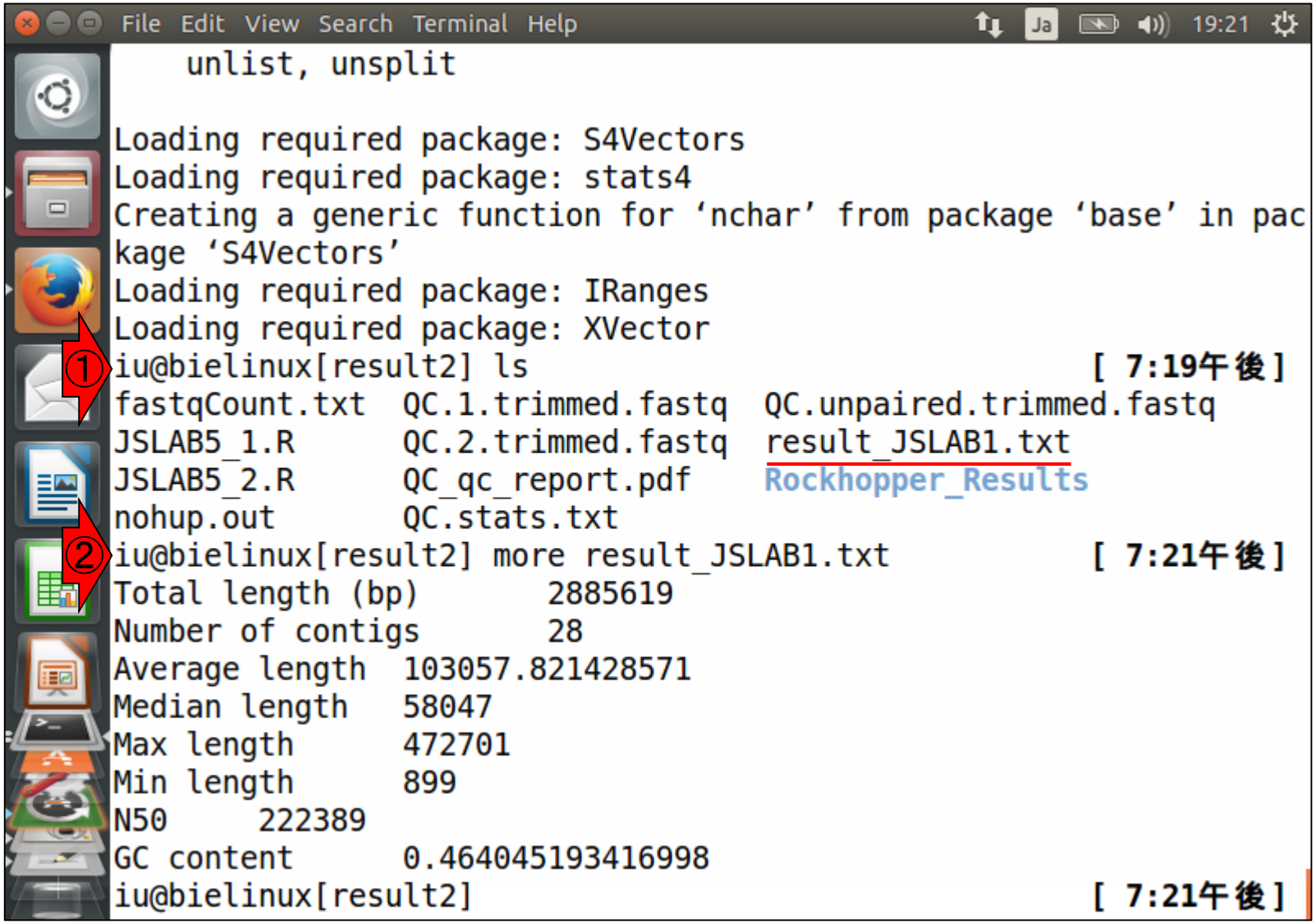
実行結果。エラーメッセージは出ていない

W12-3：発展形3



W12-3：発展形3

①lsで確認。確かに出力ファイルが存在し、②その中身も正しい

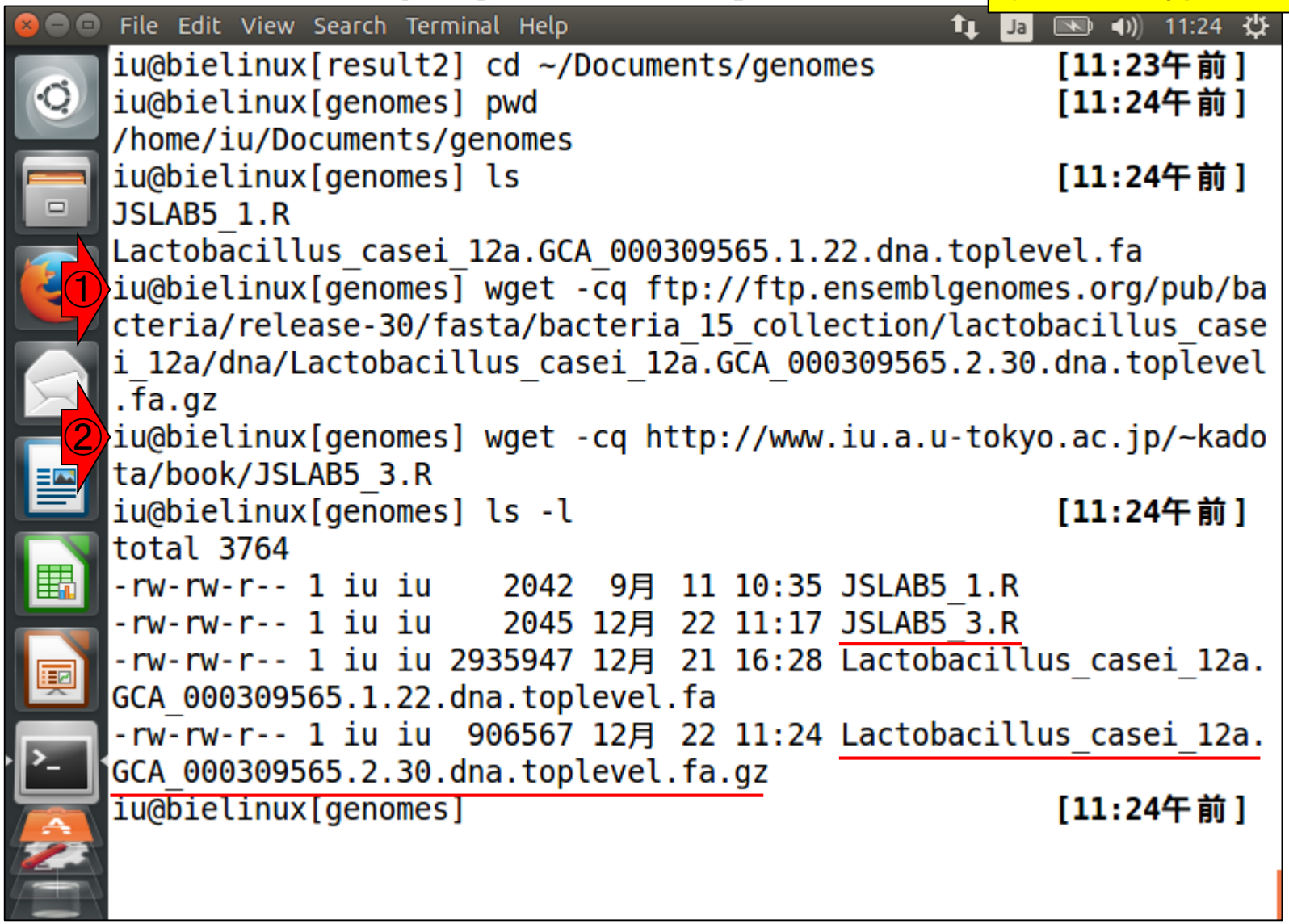


Contents

- 日本乳酸菌学会誌のNGS連載第4回までの復習（特にFastQCとFaQCs）
  - まずはFaQCs実行、おさらい、FastQCでIllumina adapterの消滅確認
- Javaプログラムの設定と実行(Rockhopper2)
  - W2: Javaの確認とダウンロード、GUI版の実行
  - W3: Linux Tips (&, ps, kill, and nohup)
  - W4とW5：コマンドライン版の実行(paired-end)、クラスパスの設定、再実行
  - W6：コマンドライン版の実行(single-end)
- Linux環境でのRの利用法
  - W7：起動と終了、QuasRパッケージのインストール（エアーハンズオン）
  - W8：R基本コマンド、W9：乳酸菌ゲノム配列取得と基本情報取得（連載第1回の図2）
  - W10：source関数、バッチモードでの利用
  - W12：バッチモードでの利用の発展形、入力ファイルの絶対パス指定
  - W13：gzip圧縮状態での利用

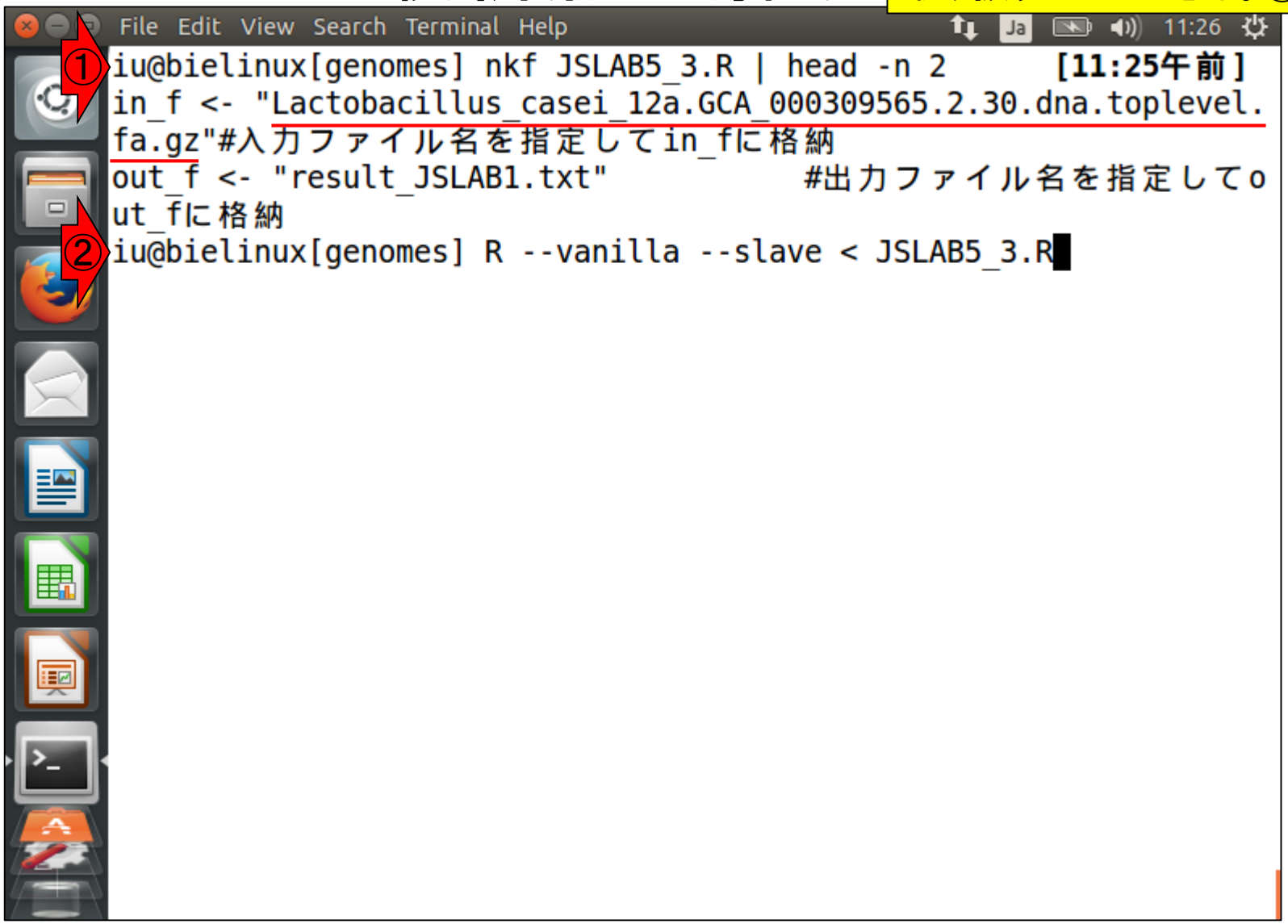
W13-1：最新版で解析

①ゲノムファイルと②Rスクリプトファイルのダウンロード。赤下線で示すように、正しく取得できてるっぽいことがわかる



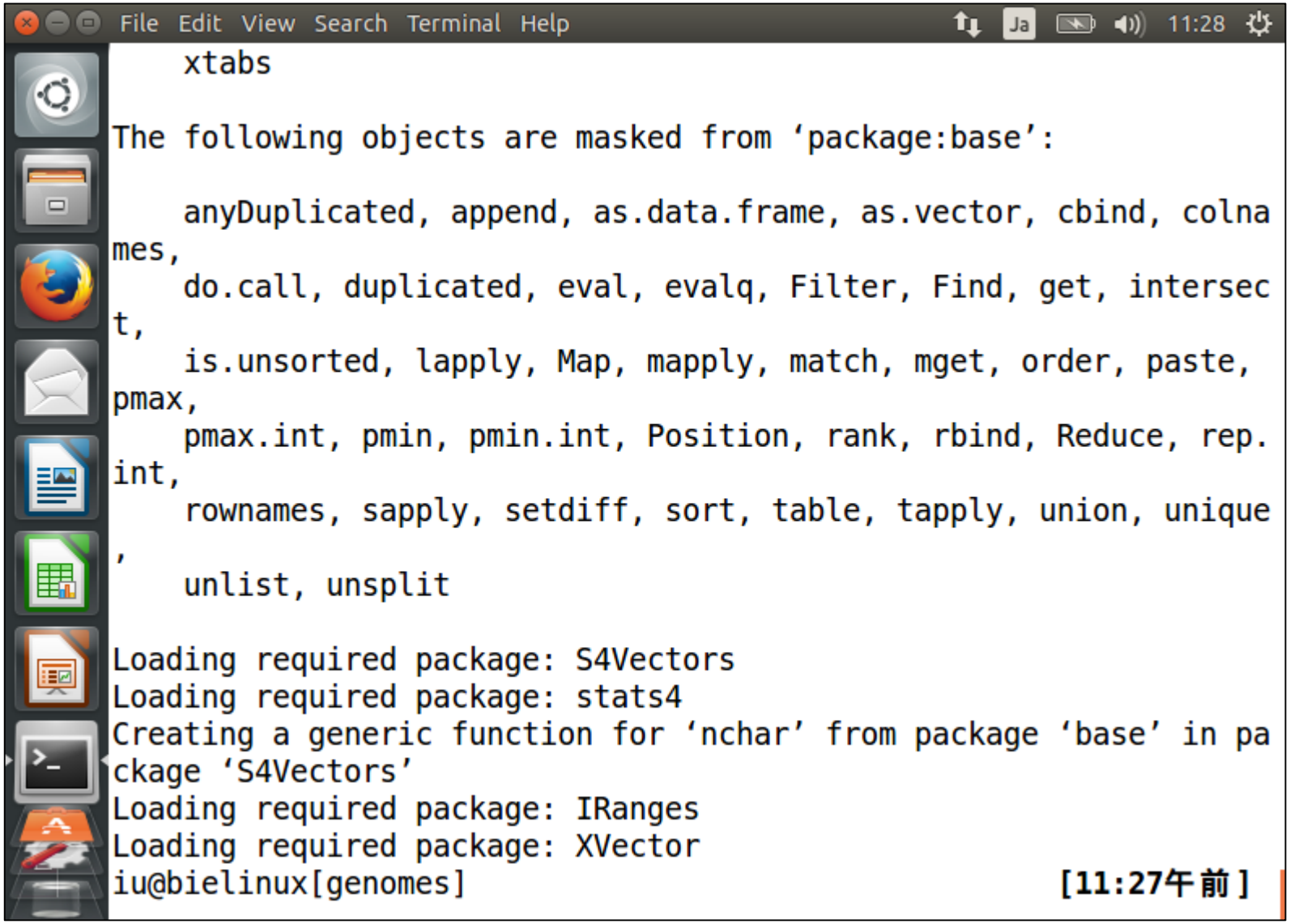
W13-1：最新版で解析

①Rスクリプトファイルの最初の2行分を表示。赤下線で示すようにgzip圧縮ファイルのままで取り扱うことができる。②JSLAB5_3.Rを実行



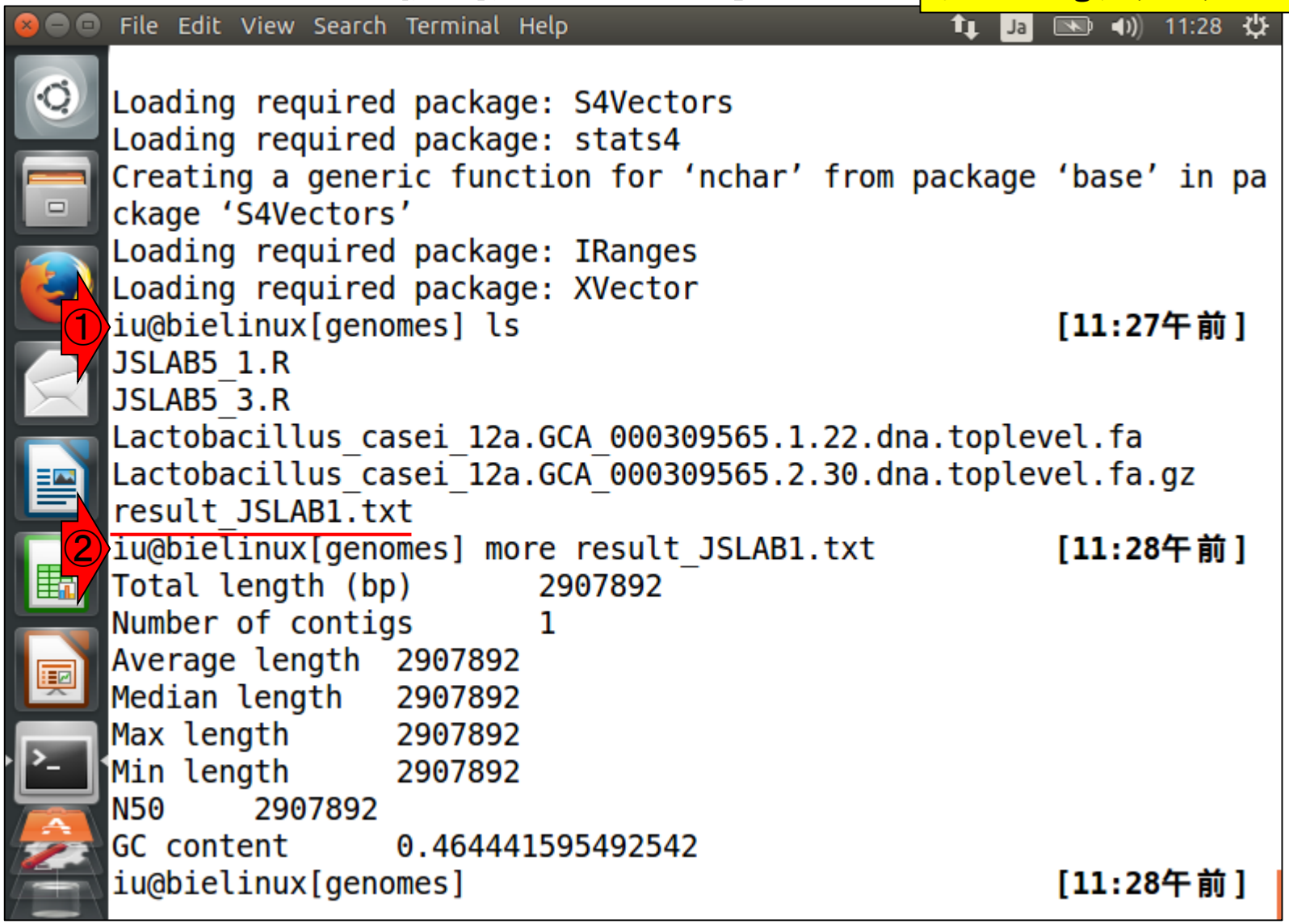
エラーなく実行できていることがわかる

W13-1：最新版で解析



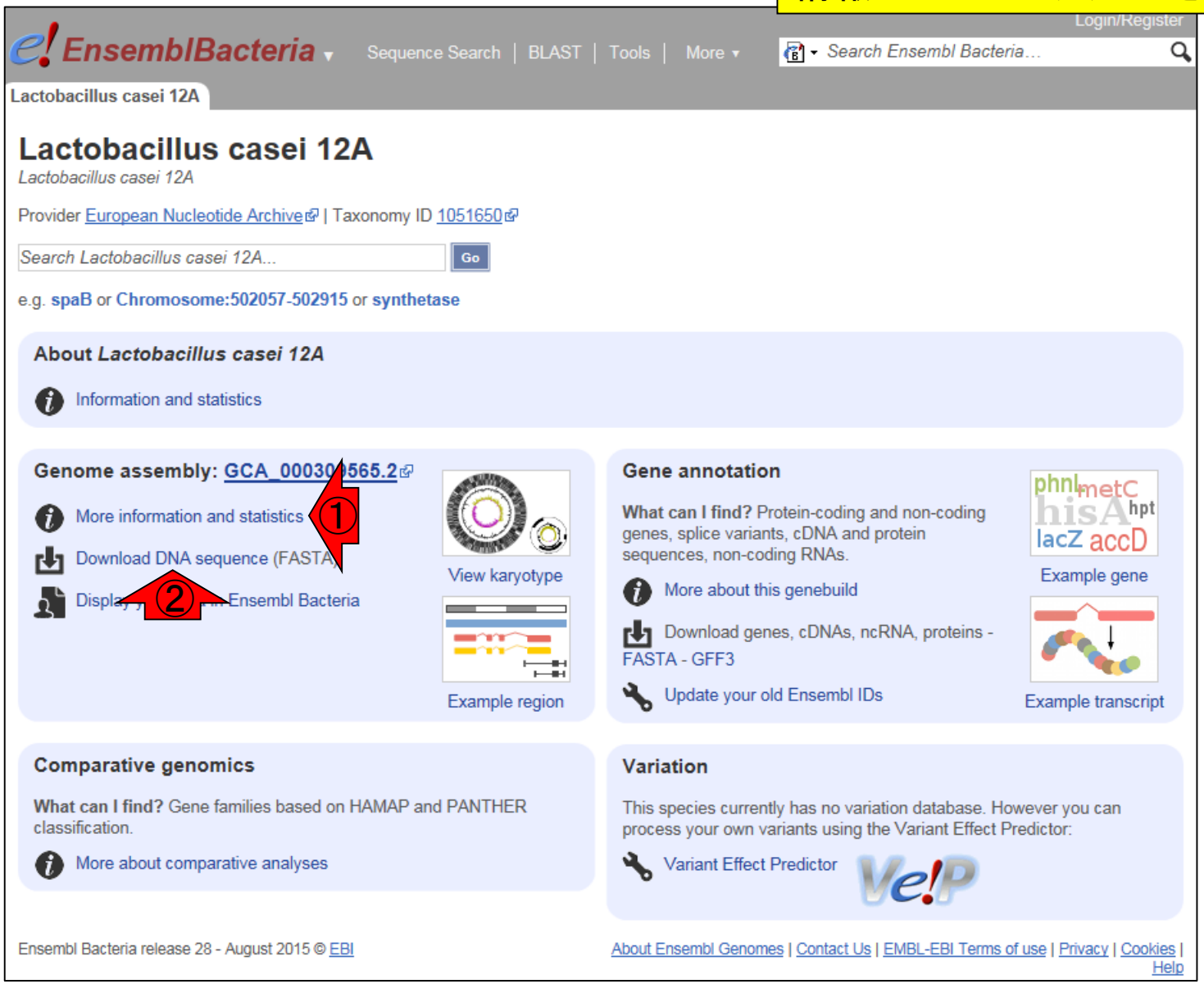
W13-1：最新版で解析

①lsで確認。出力ファイルresult_JSLAB1.txtが確かにできている。②moreで中身を表示。1 contig、2,907,892 bpであることがわかる



W13-2：Ensembl

①*L. casei* 12Aの詳細情報はここからみられる。②wgetでgzip圧縮FASTA形式ファイル取得する際のURL情報はここからゲットできます。スライドを見るだけ



W13-2：Ensembl

①をクリックした結果。連載第1回当時はコンティグレベルだったが、第5回執筆時には②染色体レベルになっていることがわかる。③トータル塩基数は2,907,892 bp

EnsemblBacteria ▾ Sequence Search | BLAST | Tools | More ▾ Search Ensembl Ba

Lactobacillus casei 12A

Lactobacillus casei 12A

Lactobacillus casei 12A

Provider European Nucleotide Archive | Taxonomy ID 1051650

Search Lactobacillus casei 12A... Go

e.g. spaB or Chromosome:502057-502915 or synthetase

About *Lactobacillus casei 12A*

Information and statistics

Genome assembly: GCA_000309565.2

More information and statistics ①

Download DNA sequence (FASTA)

Display your data in Ensembl Bacteria

View karyotyp

Example regio

Comparative genomics

What can I find? Gene families based on HAMAP and PANTHER classification.

More about comparative analyses

Ensembl Bacteria release 28 - August 2015 © EBI

Lactobacillus casei 12A Assembly and Gene Annotation

Lactobacillus casei 12A

Organism

Taxonomy ID	1051650
Name	*Lactobacillus casei 12A*
	Wikipedia
Aliases	Lactobacillus casei str. 12A Lactobacillus casei strain 12A
Classification	› cellular organisms › Bacteria › Firmicutes › Bacilli › Lactobacillales › Lactobacillaceae › Lactobacillus › Lactobacillus casei group › Lactobacillus casei › Lactobacillus casei 12A

European Nucleotide Archive Records

CP006690.1

References

1. Analysis of the Lactobacillus casei supragenome and its influence in species evolution and lifestyle adaptation. Broadbent J.R., Neeno-Eckwall E.C., Stahl B., Tandee K., Cai H., Morovic W., Horvath P., Heidenreich J., Perna N.T., Barrangou R., Steele J.L. - *BMC Genomics* 2012, **13**:533 PubMed: 23035691

Ensembl Genomes API Example

This example Perl script shows how to create a database adaptor for this species. For more information see the Ensembl Bacteria documentation

Statistics

Summary

Assembly:	ASM30956v2, INSDC Assembly GCA_000309565.2,
Database version:	81.1
Base Pairs:	2,907,892 ③
Golden Path Length:	2,907,892 ③
Data source:	European Nucleotide Archive
Genebuild method:	Generated from ENA annotation

Gene counts

Coding genes:	2,681
Non coding genes:	72
Small non coding genes:	72
Pseudogenes:	46
Gene transcripts:	2,799

Coordinate Systems

chromosome 1 sequence

Filter

Sequence	Length (bp)
Chromosome	2907892

②