

新学術領域『生命科学系4分野支援活動』ゲノム支援
平成25年度 情報解析講習会

Linux における
ソフトウェアのインストール

東京大学大学院新領域創成科学研究科
情報生命科学専攻 笠原 雅弘

ソフトウェア配布の2大形態

～ソース配布とバイナリ配布～

ソースコード配布

- ・人間が直接読むことのできるテキスト。機械が直接実行することはできない。
- ・OSの種類やCPUの種類に依らず、多くの場合1種類のファイルを配布。
- ・「コンパイラ」と呼ばれるソフトウェアでバイナリに変換(コンパイルと呼ぶ)。

バイナリ配布

- ・機械が直接解釈する16進値の羅列。極一部の達人を除き人間は読めない。
- ・OSの種類やCPUの種類毎に異なるバイナリが必要になることが多い。
- ・コンパイルが必要無いため、インストールは比較的短時間で行える。

```
void do_composition(int argc, char** argv) {
    bool flag_ignore_case = false;
    bool flag_only_monomer = false;
    bool flag_only_bimer = false;
    bool flag_only_trimer = false;
    bool flag_dapi_check = false;
    static struct option long_options[] = {
        {"ignorecase", no_argument, 0, 'i'},
        {"monomer", no_argument, 0, '1'},
        {"bimer", no_argument, 0, '2'},
        {"trimer", no_argument, 0, '3'},
        {"dapicheck", no_argument, 0, 'd'},
        {0, 0, 0, 0} // end of long options
    };
    while(true) {
        int option_index = 0;
        int c = getopt_long(argc, argv, "", long_options, &option_index);
        if(c == -1) break;
        switch(c) {
            case 0: break;
            case 'i':
                flag_ignore_case = true;
```

```
0252240 e8 43 df fe ff 48 8d bc 24 40 04 00 00 e8 36 df
0252260 fe ff e9 3f fd ff ff 8b 47 10 8d 50 ff 89 57 10
0252300 eb 85 8b 47 10 8d 50 ff 89 57 10 eb a6 49 8d 7c
0252320 24 10 48 89 c5 e8 0e df fe ff 49 8d 7c 24 08 e8
0252340 04 df fe ff eb b2 48 89 c5 eb ef 48 89 c5 66 90
0252360 eb a6 48 89 c5 eb ae e9 29 fd ff ff 48 89 c5 48
0252400 8d bc 24 10 04 00 00 e8 dc de fe ff 48 8d bc 24
0252420 20 04 00 00 e8 cf de fe ff e9 d8 fc ff ff 48 89
0252440 c5 eb e9 8b 47 10 8d 50 ff 89 57 10 e9 b6 f2 ff
0252460 ff b8 98 36 40 00 48 85 c0 74 2f 83 c8 ff f0 0f
0252500 c1 47 10 85 c0 0f 8f 6e ee ff ff 48 8d b4 24 78
0252520 04 00 00 e8 d0 de fe ff e9 5c ee ff ff 4c 89 e7
0252540 48 89 c5 e8 58 d0 ff ff eb 95 8b 47 10 8d 50 ff
0252560 89 57 10 eb ce 49 8d 7c 24 10 48 89 c5 e8 66 de
0252600 fe ff 49 8d 7c 24 08 e8 5c de fe ff e9 6e ff ff
0252620 ff 48 89 c5 eb ec 90 90 90 90 90 90 90 90 90
0252640 41 57 4c 8d 5f 10 41 56 4d 89 da 41 55 49 89 f5
0252660 41 54 55 48 89 fd 53 48 89 d3 48 81 ec 68 04 00
0252700 00 48 8b 47 18 48 85 c0 74 6c 4c 8b 0a 4d 8b 41
0252720 e8 eb 1a 0f 1f 44 00 00 89 d1 85 c9 78 4f 48 8b
0252740 50 10 49 89 c2 48 85 d2 74 4c 48 89 d0 48 8b 70
0252760 20 4c 89 cf 48 8b 56 e8 49 39 d0 48 89 d1 49 0f
```

ソースコードによる配布と バイナリ配布の損と得

ソースコードを配布

得:

(ソフトウェア作者が)様々なOSやCPUの種類に応じて個別のファイルを配布しなくて済むため、手間が少ない。
(ユーザーが)プログラムの中身を解析できるため、プログラミングが得意であれば不具合を自分で修正できるし、動作を完全に把握することができる。

損:

(ソフトウェア作者が)秘密を隠すことは難しいため、商用ソフトに適さない。
(ユーザーが)コンパイルしてバイナリに変換しなければソフトウェアを使うことができず、余計な手間が掛かる。

バイナリを配布

得:

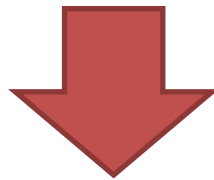
(ソフトウェア作者が)ソースコードを隠すことができ、商用ソフト等に適している。
(ユーザーが)コンパイルする必要がなく、ソフトウェアの導入に掛かる時間が短い。

損:

(ソフトウェア作者が)個別に多くの種類のファイルを配布しなくてはならず手間が大きい。
(ユーザーが)プログラムの中身を解析することができないため、動作を完全に把握することが難しい。不具合の修正も自分では行うことができない。

ソースコードによる配布と バイナリ配布の損と得（結論）

「Linux もプログラミングも初心者」という人にとって
ソースコード配布のメリットは皆無に近い。



自分のOSやCPUの種類に適合するバイナリが
配布されている場合にはバイナリを使うべし。

ソースコードを使うのは、バイナリが配布されて
いないときだけにするのが良い。

Windows/Mac と Unix (含Linux) のソフトウェア配布における文化の違い

Linux 等 (*NIX)

- ・サポートする OS や CPU の種類が幅広いためソースコードを配布することが多い。
- ・バイナリ配布を行う場合も、Linuxカーネルのバージョンや OS の種類、CPUの種類に応じて異なるファイルが配布される。
- ・自由な(フリーな)ソフトウェアの多くでは、ソフトウェアの開発者とは異なる有志がソースコードをコンパイルしてバイナリパッケージを作り、配布していることが多い。

Windows / Mac OS X

- ・UNIX上でも開発されているソフトウェアを除いてバイナリで配布されることが多い。
- ・バイナリも通常、32bit用/64bit用の2種類しか存在しない。
- ・商用ソフトの比率が高く、ソフトウェアの提供元がバイナリ配布を行っている。

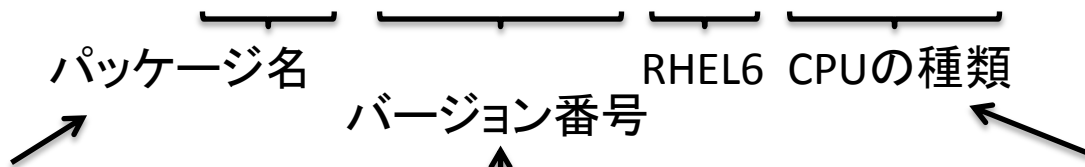
ゲノム解析分野ではこちらの文化が主流

RPM: RPM Package Manager

- RedHat 社が提供する Linux ディストリビューションおよびその派生ディストリビューションで用いられているパッケージマネージャ
 - より具体的には RedHat Enterprise Linux, CentOS, Scientific Linux, Fedora Core, Vine Linux などで用いられている。

RPMファイルの
命名規則:

```
[root@zombie tmp]# ls *.rpm  
mosh-1.2.2-1.el6.x86_64.rpm
```



基本的にはソフトウェアの名前をそのまま用いる。ライブラリは“lib****”としたり、コンパイルを行う時にのみ必要なパッケージは“****-devel”と名付けることになっている。

基本的には各ソフトウェアのバージョンをそのまま用いる。細かい修正を入れた場合に末尾が変わる。

x86_64 Intel/AMD の 64 bit CPU
i386~ Intel/AMD の 32 bit CPU
i686 (64bit CPU でもメモリーが 32bit CPU 並にしか使えないだけで動作はする)
sparc 京コンピュータなど
ppc POWER (IBM製)
noarch CPUの種類に依存しない

よく使うRPMコマンド

インストール

```
[root@zombie tmp]# rpm -ivh mosh-1.2.2-1.el6.x86_64.rpm
```

バージョンアップ

```
[root@zombie tmp]# rpm -Uvh mosh-1.2.4-1.el6.x86_64.rpm
```

削除

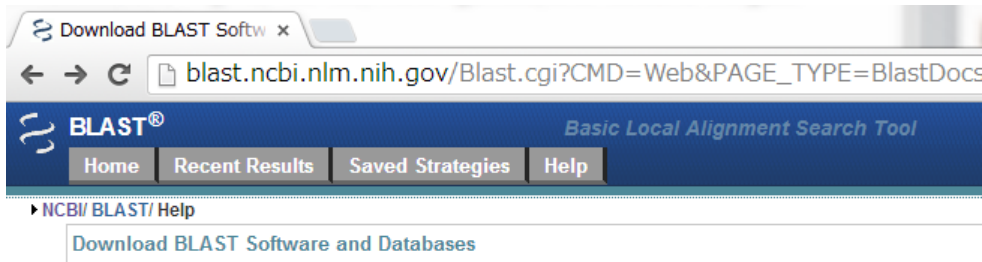
```
[root@zombie tmp]# rpm -e mosh
```

え〜っと、あのパッケージ入ってたっけか？

```
[root@zombie tmp]# rpm -qa | grep -i mosh  
mosh-1.2.2-1.el6.x86_64
```

- ※ rpm を使ったソフトウェアのインストールには管理者権限が必要。
遺伝研スパコンなど、ユーザー権限のみ持っている場合には使えない。
rpm パッケージがある場合、管理が楽なので管理者にお願いしてもそれほど大きな負担にはならない。

NCBI-BLAST+ の場合



BLAST+ executables

BLAST+ is a new suite of BLAST tools that utilizes the NCBI C++ Toolkit. The BLAST+ applications have been described in the article in BMC Bioinformatics ([PubMed link](#)).

Installers and source code are available from [ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/](http://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/).

blast.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/

blast/executables/blast+/LATEST/ のインデックス

名前	サイズ	更新日
[親ディレクトリ]		
ChangeLog	12.1 kB	13/04/01 0:00:00
README.solaris	178 kB	13/06/04 19:20:00
ncbi-blast-2.2.28+-1.i686.rpm	160 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-1.src.rpm	10.8 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-1.x86_64.rpm	137 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-ia32-linux.tar.gz	160 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-ia32-win32.tar.gz	54.2 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-src.tar.gz	12.8 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-src.zip	15.7 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-universal-macosx.tar.gz	260 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-win32.exe	54.3 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-win64.exe	62.1 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-x64-linux.tar.gz	137 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+-x64-win64.tar.gz	61.9 MB	13/04/01 0:00:00
ncbi-blast-2.2.28+.dmg	261 MB	13/04/01 0:00:00

rpm/パッケージ

※ .src.rpm という拡張子のファイルは“ソースrpm”と呼ばれ、「バイナリではない」ので初心者には不向き。

yum

- rpm パッケージのダウンロードとインストールを自動化するソフトウェア

インストール

```
[root@zombie tmp]# yum install mosh
```

アップデート(バージョンアップ)

```
[root@zombie tmp]# yum update mosh
```

全部アップデート

```
[root@zombie tmp]# yum update
```

パッケージのサーチ

```
[root@zombie tmp]# yum search mosh
```

多くの Linux ディストリビューションでは、多くのパッケージを集めた「リポジトリ」と呼ばれるサーバが用意されている。

例) CentOS 用の rpm リポジトリサーバ

2) mosh の rpm をダウンロード

1) yum install mosh

yum

3) rpm -ivh でインストール

rpm

※rpm を使っているので管理者権限が必要な事に注意。
必要なら sudo や su コマンドを用いて管理者権限で作業する。

Dependency (依存関係)

- AAA というパッケージを入れるためには BBB というパッケージを入れる必要がある。
- BBB というパッケージを入れるためには CCC というパッケージを入れる必要がある。
- ...

“yum install AAA” コマンドを用いる場合には、リポジトリ中から BBB や CCC も探し出して同時にインストールしてくれるので非常に便利。リポジトリ中に、AAAのインストールに必要なパッケージが足りない場合にはエラーとなる。

RedHat 系列以外の場合には yum の代わりに apt-get, zypper , emerge, pacman 等、yum と同様のパッケージマネージャが用意されていることが多い。

Bioinformatics のソフトウェア達を インストールする際の問題点

- rpm パッケージのように便利な手段が使えることはほとんど無い
 - NCBI-BLAST のように、大手が開発しているソフトウェアではまれに rpm で提供されている。
 - 単独の研究室で作っているソフトウェアは rpm で提供されていることはほぼあり得ない。
- Dependency Hell
 - rpm が無い、ということはソフトウェアの依存関係を自分で処理する必要がある。
 - ソフトウェアを一つインストールするためだけに、そのソフトウェアの動作に必要な10以上のソフトウェアを入れる必要があることも珍しくない。
 - 依存関係が複雑なソフトウェアはインストールに丸1日掛かっても不思議ではない。

ソースコードからのインストール

- ソースコードのダウンロード
– 例) SAMtools


Home / Browse / Science & Engineering / Bio-Informatics / SAM tools / Files



SAM tools

Brought to you by: [bhandsaker](#), [jmarshall](#), [lh3lh3](#), [petulda](#)

Summary | **Files** | Reviews | Support | Wiki | Mailing Lists | Hosted Apps ▾ | Code

Looking for the latest version? [Download samtools-0.1.19.tar.bz2 \(514.5 kB\)](#)

Home / [samtools](#) / 0.1.19 

Name ↕	Modified ↕	Size ↕	Downloads / Week ↕
↑ Parent folder			
samtools-0.1.19.tar.bz2	2013-08-19	514.5 kB	1,066  

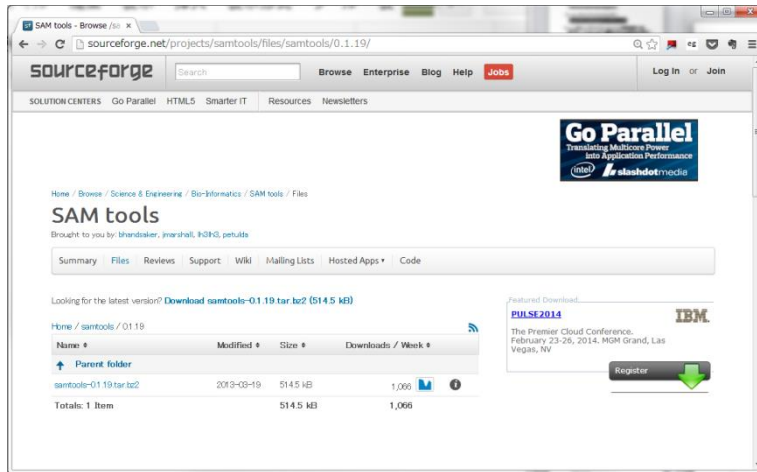
Totals: 1 Item

Click to download samtools-0.1.19.tar.bz2 514.5 kB 1,066

ソースコードは “.tar.gz”, “.tar.bz2”, “.tar.xz” など、tar を用いて複数のファイルをまとめた「tar アーカイブ」で配布するのが一般的。“tar ball” と呼ばれる。

ソースコードをダウンロード

- (1)手元の端末(Win/Mac/Linux/...)で SAMtools の web ページを開く
- (2)右クリック等のメニューから「リンクをコピー」する。



- (3) TeraTerm (右クリック), PuTTY (右クリック), Terminal.app (CMD+V) 等の端末にアドレスをコピーしてファイルをダウンロードする。

```
[mkasa@t352] ~% wget http://sourceforge.net/projects/samtools/files/samtools/0.1.19/samtools-0.1.19.tar.bz2/download
```

wget はダウンロードするソフト

‘?’ や ‘&’ が含まれるURLは ‘’ で括らないと
“no matches” エラーが発生するので注意。

tarball を解凍する

- “tarball” には複数のファイルが格納されているので「解凍」と呼ばれる操作でファイルを取り出す。

tar.gz の場合

```
% tar xvfz samtools-0.1.19.tar.gz
```

tar.bz2 の場合

```
% tar xvfj samtools-0.1.19.tar.bz2
```

付属の説明書を読む

- たいていのソフトウェアには「説明書」が付属している。「説明書」は面倒でも必ず読むべし。

INSTALL や Install.txt、
README、README.txt など、
ソフトウェアをインストールし
て使うために必要な手順が書
かれたファイルは熟読しよう。

NEWS/AUTHORS/COPYING/Ch
angeLog などは読まずとも問
題は少ない。ChangeLog には
変更履歴が書かれているの
で、バージョンアップで問題が
解決しているか等を知る手が
かりになる。

```
[mkasa@t262]~% cd samtools-0.1.19
[mkasa@t262]~/samtools-0.1.19% ls
AUTHORS          bam_index.c      bedidx.c         ksort.h
COPYING          bam_lpiledup.c   bgzf.c           kstring.c
ChangeLog.old    bam_mate.c       bgzf.h           kstring.h
INSTALL          bam_md.c         bgzip.c          misc
Makefile         bam_pileup.c     cut_target.c     padding.c
Makefile.mingw   bam_plcmd.c     errmod.c         phase.c
NEWS             bam_reheader.c  errmod.h        razf.c
bam.c            bam_rmdup.c     examples        razf.h
bam.h            bam_rmdupse.c   faidx.c         razip.c
bam.h~          bam_sort.c      faidx.h         sam.c
bam2bcf.c        bam_stat.c      kaln.c          sam.h
bam2bcf.h        bam_tview.c     kaln.h         sam_header.c
bam2bcf_indel.c bam_tview.h     khash.h        sam_header.h
bam2depth.c      bam_tview_curses.c klist.h        sam_view.c
bam_aux.c        bam_tview_html.c knetfile.c     sample.c
bam_cat.c        bamshuf.c       knetfile.h     sample.h
bam_color.c     bamtk.c         kprobaln.c    samtools.1
bam_endian.h    bcftools        kprobaln.h    win32
bam_import.c    bedcov.c        kseq.h
```

Autotools を用いて作成された tarball のインストール手順

- tarball に “configure” というファイルが含まれていれば autotools を用いて作成されている（正しくは、autotools を用いて作成されたと見なして操作して構わない。）
- tarball をダウンロード (k-mer 解析用ソフト Jellyfish の例)

```
% wget 'http://www.cbcb.umd.edu/software/jellyfish/jellyfish-1.1.11.tar.gz'
```

- 解凍し、ファイルが存在するディレクトリに移動

```
% tar xvfz jellyfish-1.1.11.tar.gz  
% cd jellyfish-1.1.11
```

- 依存ライブラリ等のチェック

```
% ./configure
```

- コンパイル

```
% make
```

- インストール

```
# make install (管理者権限以外で操作しているときには sudo make install)
```


/home の下にソフトウェアをインストールする

- ソフトウェアのインストールには基本的に管理者権限が必要となる。
- Autotools で作られた tarball は管理者権限の要らないホームディレクトリの下へインストールすることが可能。それ以外の tarball は説明書を読むべし。

具体的なやり方(前のスライドの “./configure” ステップだけ変更する)

```
% ./configure --prefix=$HOME/local
```

“\$HOME” は自分のホームディレクトリ名に置換されるので、上のコマンドは以下と同じ意味になる。(アカウント名が mkasa の場合)

```
% ./configure --prefix=/home/mkasa/local
```

実行ファイル(ソフトウェアの本体)は /home/mkasa/local/bin の下に通常は配置される。(アカウント名が mkasa の場合)

LPM: Local Package Manager

← → ↻ www.kasahara.ws/lpm/ 🔍 ☆ 🇯🇵 e

LPM: Local Package Manager Want to install software without root?

[Home](#)

[English](#) | [Japanese](#)

Home

Do you have an experience of installing software without root privilege on Linux? Have you struggled with software that wasn't packaged well? Do you have a custom patch for a package to avoid a compiler error on your environment? LPM might be what you need.

Navigation

- [Home](#)
- [Introduction](#)
- [Download](#)
- [Quick Start](#)
- [Installation](#)
- [Document](#)
- [Browse Packages](#)
- [Submit New Script](#)
- [Change Log](#)

Design by [Minimalistic Design](#)



DBCLS の Open Science Award 2013 の
審査員特別賞に選ばれました。

LPM の開発経緯

- ゲノム解析分野で使うソフトウェアは、そのほとんどがきちんとバイナリパッケージ化されていない。
- 小さい研究室ではスパコンを借りたり、コラボレーターのクラスター計算機を借りることが多い。
- 計算機を借りるたびにインストールからやり直し。
- ソフトウェアのインストールで丸1週間潰れるのも普通だった。



- きちんとバイナリパッケージ化されていないソフトを自動でインストールできないと研究が進まない。
- LPM を開発してゲノム解析のソフトを1コマンドでインストール。

LPM のインストール

さっさと使ってみたい

LPM 自身のインストール。

```
$ cd
$ wget http://www.kasahara.ws/lpm/lpm
$ chmod +x lpm
$ ./lpm initlocaldir
$ rm ./lpm
```

インストールが終わったら再ログインしてください。面倒なら以下のコマンドを実行すれば良いでしょう。

```
$ exec $SHELL -l
```

さて、例えば **ttyrec** をインストールしてみましょう。

```
$ ttyrec
bash: ttyrec: Command not found.
$ lpm install ttyrec
(いろいろ表示される)
$ $SHELL -l (あるいは再ログイン)
$ ttyrec -h
usage: ttyrec [-u] [-e command] [-a] [file]
$
```

インストールされているソフト一覧表示。

```
$ lpm list
 2M   1  02-Feb-2010  lpm-1.0
28M  13  06-Sep-2010  maven-2.0.11
22M  13  02-Feb-2010  paco-2.0.7
12M   6  06-Sep-2010  ttyrec-1.0.8
```

特定パッケージのファイル一覧表示。

```
$ lpm list maven
```

アンインストール。

```
$ lpm uninstall maven
```

時間があればここで Jellyfish のインストールをやってみます。

```
% lpm install jellyfish
```

2分ほどでコンパイルとインストールが終わります。

```
% lpm install jellyfish
```

このようになれば完了です。

```
[mkasa@k03]~% jellyfish
Too few arguments
Usage: jellyfish <cmd> [options] arg...
Where <cmd> is one of: count, stats, histo,
Options:
--version          Display version
--help             Display this message
```

そもそも LPM 用のパッケージが無いので **tarball** からいきなりインストール

```
$ lpm install http://tukaani.org/xz/xz-4.999.9beta.tar.bz2
```

データベースが古いよ、と警告メッセージがたまに

```
$ paco -ua
```

ほとんどの場合はこの説明の使い方で十分なはず。

バイオインフォマティクス関連だけでも100以上のパッケージが用意されています。

Available packages in the default public repository.

パッケージ名	ver	説明	ホームページ	備考	テスト環境	依存パッケージ
abyss-mpi	1.3.6	ABYSS assembler (MPI)	CentOS 6.4			link
abyss	1.3.6	ABYSS assembler (Non-MPI)	CentOS 6.4			link
allpaths-lg	47503	ALLPATHS-LG Genome Assembler	CentOS 6.4			link
amos	3.1.0	AMOS is a collection of tools and class interfaces for the assembly of DNA reads.	SL6.1			link
ant	1.9.2	ant	CentOS 6.4			link
arachne	3.2_March9_2009	Arachne assembler	SL6.1			link
autosh	1.4c	autosh				link
bam2fastq	1.1.0	The BAM format is an efficient method for storing and sharing data from modern, highly parallel sequencers. While primarily used for storing alignment information, BAMs can (and frequently do) store unaligned reads as well.	SL6.1			link
bamttools		BAMtools is a project that provides both a C++ API and a command-line toolkit for reading.	SL6.1			link cmake
bedtools	2.17.0	BEDTools is an extensive suite of utilities for comparing genomic features in BED format.	CentOS 6.4			link
bfast	0.7.0	BFAST Blat-like Fast Accurate Search Tool	SL6.1			link
biopython	1.56	BioPython	SL6.1			link
biotoolbox	1.12.3	biotoolbox	CentOS 6.4			link
blasr	20130820	BLASR Basic Local Alignment with Successive Refinement	CentOS 6.4			link
blast+	2.2.26+	NCBI BLAST+	CentOS 6.4			link

メーリングリスト:

“LPM Users ML (JP)” で検索

“LPM Users ML” (英語) もあります。

LPMスクリプトの投稿をしてくれると嬉しいですが、新たなソフトウェアへの対応リクエストだけでも歓迎です。