バイオインフォマティクスセンターの計算機システムの概要と利用方法

http://www.gen-info.osaka-u.ac.jp/

後藤　直久

2023年9月12日

バイオインフォマティクスセンター計算機システム

- 遺伝子・ゲノム・NGS情報解析用コンピュータ
- 全学共同利用
  - 登録者数：約300名
- レンタル・5年間（政府調達の入札）
  - 2022年3月機器更新
    - 計算サーバー： 384CPUコア・メインメモリ12TB
    - ファイルサーバー：実効容量4PB

計算機システム沿革

- （1991「遺伝子教育実験施設」（仮）設置準備委員会設立）
- （1992「遺伝情報実験施設」設置（独立部局））
- 1994 計算機システム導入
- （1995 南館改修工事完了・施設本格稼働）
- 1998 計算機システム更新
- （2001/4「遺伝情報実験センター」に改組）
- 2002/3 計算機システム更新
- （2004/4 国立大学法人化）
- （2005/4 微生物病研究所附属に改組）
- 2007/3 計算機システム更新
- 2012/3 計算機システム更新
- 2017/3 計算機システム更新
- 2022/3 計算機システム更新
- （2023/5「バイオインフォマティクスセンター」に改組）

4年

4年

5年

5年

5年

5年

落札額

期間	税別月額	消費税	税込月額	年額(税込)	総額(税込)
2007/3-2012/2		5%			
2012/3-2017/2	4,650,000	5%	4,882,500	58,590,000	292,950,000
	↓25%縮減				
2017/3-2022/2	3,464,000	8%	3,741,120	44,893,440	224,467,200
	↓10%縮減				
2022/3-2027/2	3,109,800	10%	3,420,780	41,049,360	205,246,800

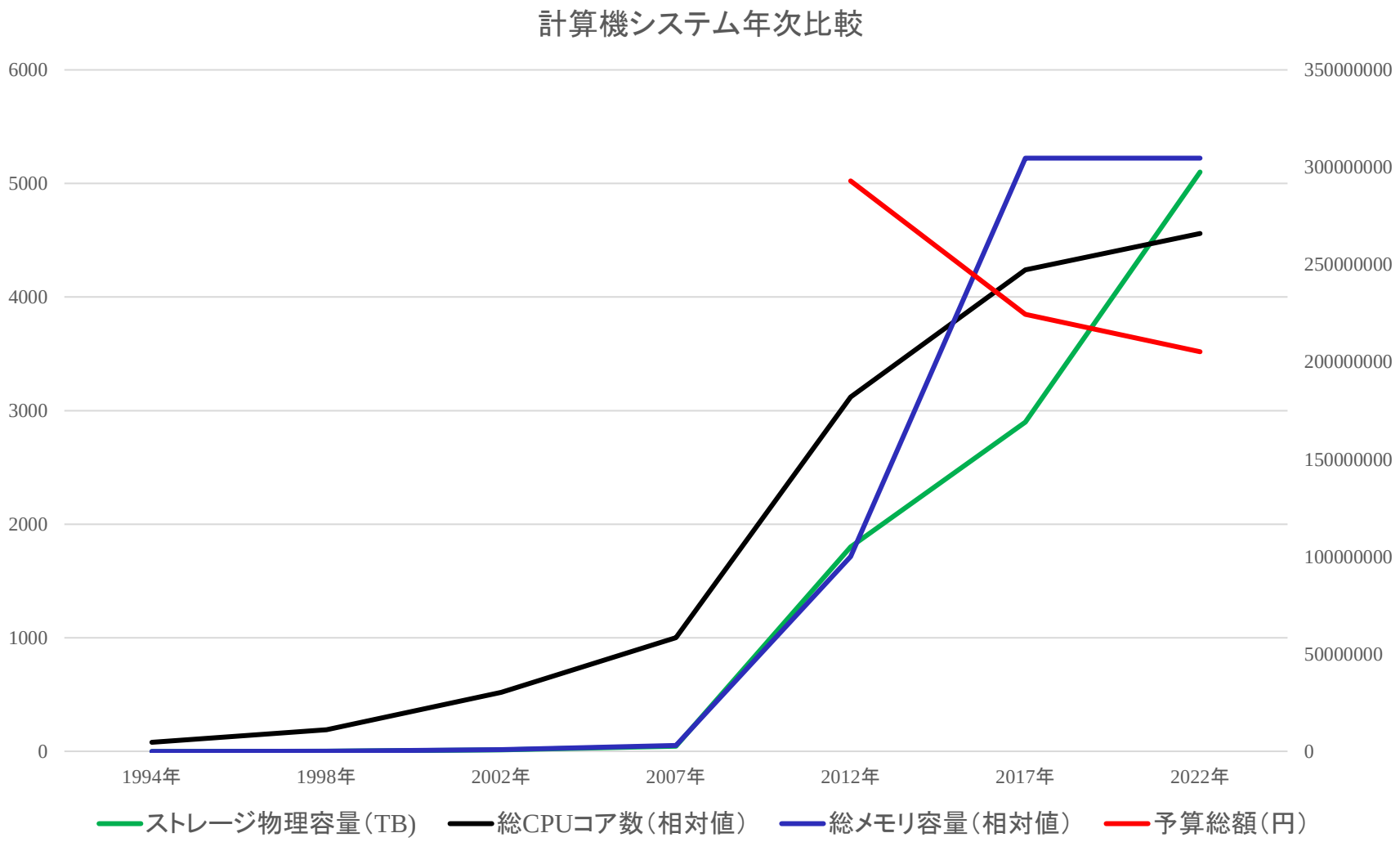
※「資産の貸付に係る経過措置」の諸条件を満たす借用(レンタル)契約のため、
契約日の消費税率が期間満了まで適用される。

主要サーバー・ストレージの変遷

oubic

年	ストレージ（物理容量）	サーバー種類	OS	CPU種類	クロック (GHz)	総コア数（コア数×CPU数×台数）	総メモリ量（容量×台数）
1994	71 GB (50GB+21GB)	共有メモリ	SunOS IRIX	SPARC MIPS	0.05(?) ?	4 (1x4) 4 (1x4)	0.25 GB (=256MB) 0.25 GB (=256MB)
1998	1.15 TB （9.1GB x 126） （実効 951 GB）	共有メモリ	Solaris Solaris	SPARC SPARC	0.25(?) 0.25(?)	15 (1x15) 4 (1x4)	5.5 GB 0.75 GB (=768MB)
2002	12 TB （73GB x 168） （実効 9.5 TB）	共有メモリ クラスタ	Solaris Solaris Linux	SPARC SPARC INTEL	0.563 0.75 1.13	16 (1x16) 4 (1x4) 32(1x2x16)	16 GB 4 GB 16 GB (1x16)
2007	44.7 TB （300GB x 149） （実効 31 TB）	共有メモリ クラスタ	Solaris Linux	SPARC INTEL	2.16 3.2	32 (1x32) 68 (2x2x17)	64 GB 68 GB (4x17)
2012	1.8 PB （3 TB x 600） （実効 1.2 PB）	共有メモリ クラスタ	Linux Solaris Linux	INTEL SPARC INTEL	2.4 2.66 3.46	80 (10x8) 16 (4x4) 216 (6x2x18)	2 TB 512 GB 1.7 TB (96x18)
2017	2.9 PB （6 TB x 490） （実効 2.7 PB）	共有メモリ	Linux Linux Solaris	INTEL INTEL SPARC	2.2 2.0 2.8	352 (22x16) 56 (14x2x2) 16 (16x1)	12 TB 512 GB (256x2) 256 GB
2022	5.1 PB （16 TB x 320） （実効 4 PB）	共有メモリ	Linux Linux	INTEL INTEL	2.1 2.2	384 (24x16) 72 (18x2x2)	12 TB 768 GB (384x2)

バイオインフォマティクスセンター計算機システム



計算機システム　サーバー機器

oubic

遺伝子解析サーバー

HPE SuperDome Flex
CPU: Intel Xeon Gold 6252 (2.1 GHz)×16
(合計384コア) (15.97 TFLOPS*)
(* AVX-512ベースコア周波数1.3GHz時の数値)
メモリ: 12 TB
OS: RedHat Enterprise Linux 8

フロントエンドサーバー

（ログイン･ジョブ管理用のサーバー）
HPE ProLiant DL360 Gen10 (2台)
CPU: Intel Xeon Gold 5220×2 (合計36コア)
メモリ: 256GB
OS: RedHat Enterprise Linux 8

ストレージ（ファイルサーバー）

DELL/EMC
PowerScale A2000 (16台) + F200 (4台)
物理容量: 5.1 PB (16TB × 320台)
実効容量: 4 PB (= 4,000 TB)

計算機システム利用方法の概要

- 事前の利用申請（ユーザー登録）が必要
- 学内ネットワーク経由のリモート利用
  - 共同利用パソコンは2022年2月末に供用終了しました
  - 初期状態では、阪大の学内ネットワークから接続可能
    - IDとパスワードによる認証
  - セキュリティの高いログイン方法を設定すれば、学外を含めた全世界から接続可能

計算機システム利用開始の流れ

申請

- 利用申請書に記入して提出
- ID・初期パスワードを返送します

接続準備

- パソコンの準備
- ソフトのインストール・設定

接続

- コマンドライン操作（SSH）
- ファイル転送（SFTP）

利用申請書

http://www.gen-info.osaka-u.ac.jp/

「利用者用ページ（学内専用）」から
PDF・Excelの書式をダウンロード

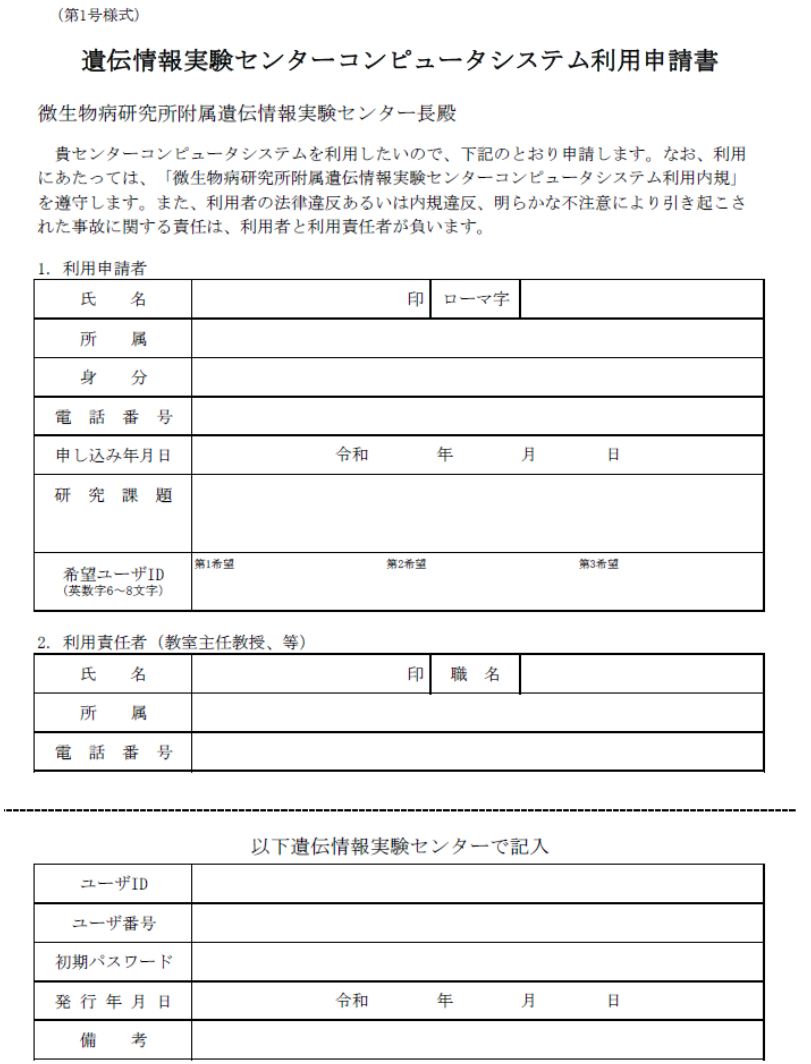
提出先：

学内便：微生物病研究所
附属バイオインフォマティクスセンター
計算機システム担当

Email：computer-system@gen-info
.osaka-u.ac.jp

（※Email提出の場合、手続の都合上、後日、原本の送付をお願いする場合があります）

（※ウェブのフォームからの申請に移行予定）



（第1号様式）

遺伝情報実験センターコンピュータシステム利用申請書

微生物病研究所附属遺伝情報実験センター長殿

貴センターコンピュータシステムを利用したいので、下記のとおり申請します。なお、利用にあたっては、「微生物病研究所附属遺伝情報実験センターコンピュータシステム利用内規」を遵守します。また、利用者の法律違反あるいは内規違反、明らかな不注意により引き起こされた事故に関する責任は、利用者と利用責任者が負います。

1. 利用申請者

氏　名	印	ローマ字	
所　属			
身　分			
電話番号			
申し込み年月日	令和　年　月　日		
研究課題			
希望ユーザID（英数字6～8文字）	第1希望　第2希望　第3希望		

2. 利用責任者（教室主任教授、等）

氏　名	印	職　名	
所　属			
電話番号			

以下遺伝情報実験センターで記入

ユーザID	
ユーザ番号	
初期パスワード	
発行年月日	令和　年　月　日
備　考	

希望ユーザIDについて

- 使用可能文字：半角英数字・一部の記号
  - アルファベット：小文字のみ
  - 数字：0～9
  - 記号：-（ハイフン）と_（アンダースコア）のみ
  - 先頭はアルファベットのみ使用可能
  - 記号の2文字連続使用は不可
  - 末尾の記号使用は不可
  - 大阪大学個人IDと類似の文字列は使用不可
- 文字数：5～15文字が目安
- 希望IDの空き状況はお問い合わせください

参考：パスワード変更方法

- ホームページ内の「利用者用ページ（学内専用）」
  http://www.gen-info.osaka-u.ac.jp/localdocs/users.html
  の「パスワード変更」から、いつでも変更可能
  - ID・現在のパスワードの入力が必要
  - 学内LANからのみ変更可能
    - （学外からパスワード変更が必要な場合は個別対応）

2022年5月以前に登録済の方へ(1)

- 新規ユーザーの「ログインシェル」が変更されました
  - 2022/4以前: /bin/csh
  - 2022/5以降: /bin/bash (現在のLinuxでは標準的)
- 既存ユーザーのログインシェルは無変更
  - 必要なら以下のページから自力で変更可能
  - 意図して /bin/csh を使用中の人以外は /bin/bash に変更が無難?

- ホームページ内の「利用者用ページ(学内専用)」
  http://www.gen-info.osaka-u.ac.jp/localdocs/users.html
  の「ログインシェル変更」から、いつでも変更可能
  - ID・パスワードの入力が必要
  - 反映に時間がかかる場合あり(最長45分間)
  - 学内LANからのみ変更可能
    - (学外から変更が必要な場合は個別対応)

2022年5月以前に登録済の方へ(2)

- ドットファイル(シェルの環境設定ファイル)の追加・修正等が必要となる場合があります
- ログインシェルを/bin/bashに変更した場合
  - ドットファイルの追加が必要:以下のコマンドを実行

```
cp -ai /etc/skel/.bash_profile ~/
```

- ログインシェルを変更せず/bin/cshを使い続ける場合
  - 一部コマンドが使えない問題が発生していたら、
    以下のコマンドを実行(問題が無ければ実行不要)

```
mv -i .cshrc .cshrc.20220301
```

- それぞれ1回のみ実行してください
- 実行に成功した場合は特に何も出力されません
  - エラー等の発生時のみメッセージが出ます

接続用ソフトのインストール

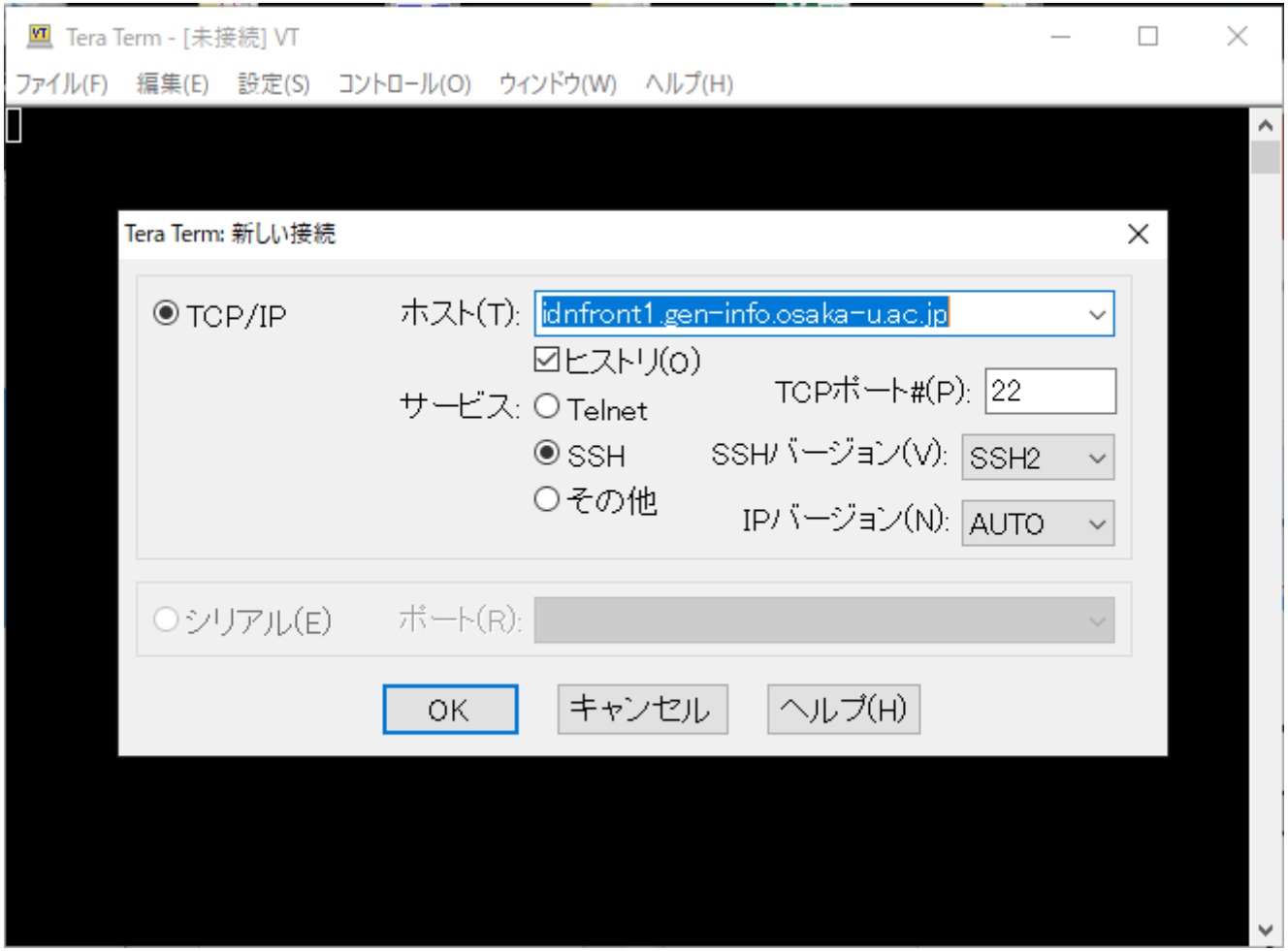
- 「SSH対応ターミナルエミュレータ」という種類のソフト
- フリーソフトウェアも多く存在
  - 無償利用可能・再配布可能・ソースコード公開
- Windows
  - Tera Term　　http://ttssh2.osdn.jp/
  - PuTTYrv　　https://www.ranvis.com/putty
- Mac
  - Terminal　　（インストール済の標準アプリ）
  - iTerm2　　https://iterm2.com/
- 全ソフト共通の注意点：原則として最新版をインストール
  - 旧バージョンは暗号化プロトコルが古く接続不可の場合があるため

Windows: Tera Term

- TeraTerm: https://ttssh2.osdn.jp/
  - Tera Term Project が開発するターミナルエミュレータ
  - フリーソフトウェア
  - ページの説明に従ってダウンロード
    - βでない最新版のインストーラをダウンロード
    - 2023年9月現在の最新バージョン：4.106 (teraterm-4.106.exe)
  - インストール
    - デフォルト選択肢のまま「標準インストール」で大丈夫

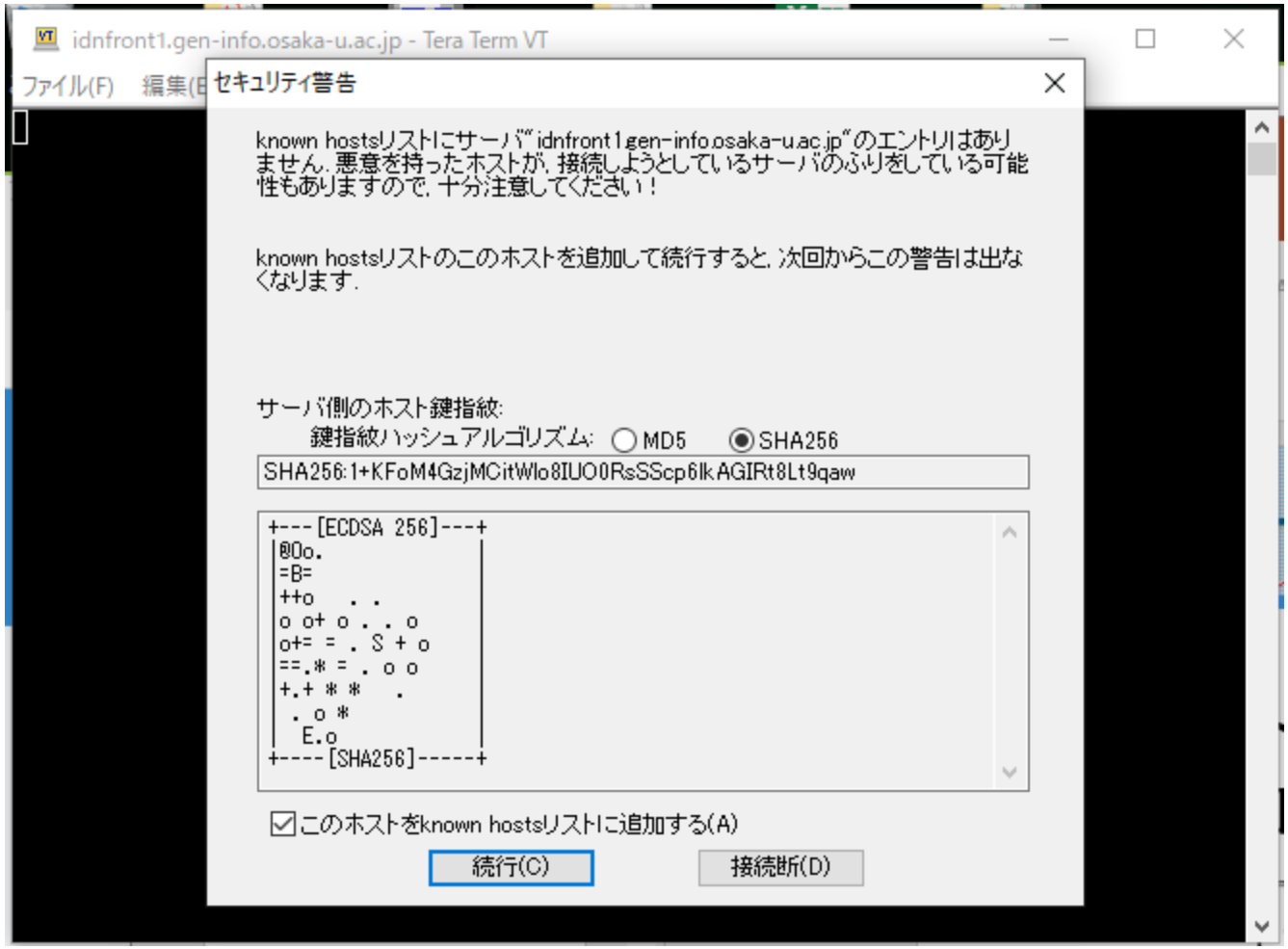
Tera Termによるサーバー接続

- 「ホスト(T)」に「idnfront1.gen-info.osaka-u.ac.jp」を入力し「OK」
  - 次回使用時は選択肢から上記ホストを選ぶだけでよい



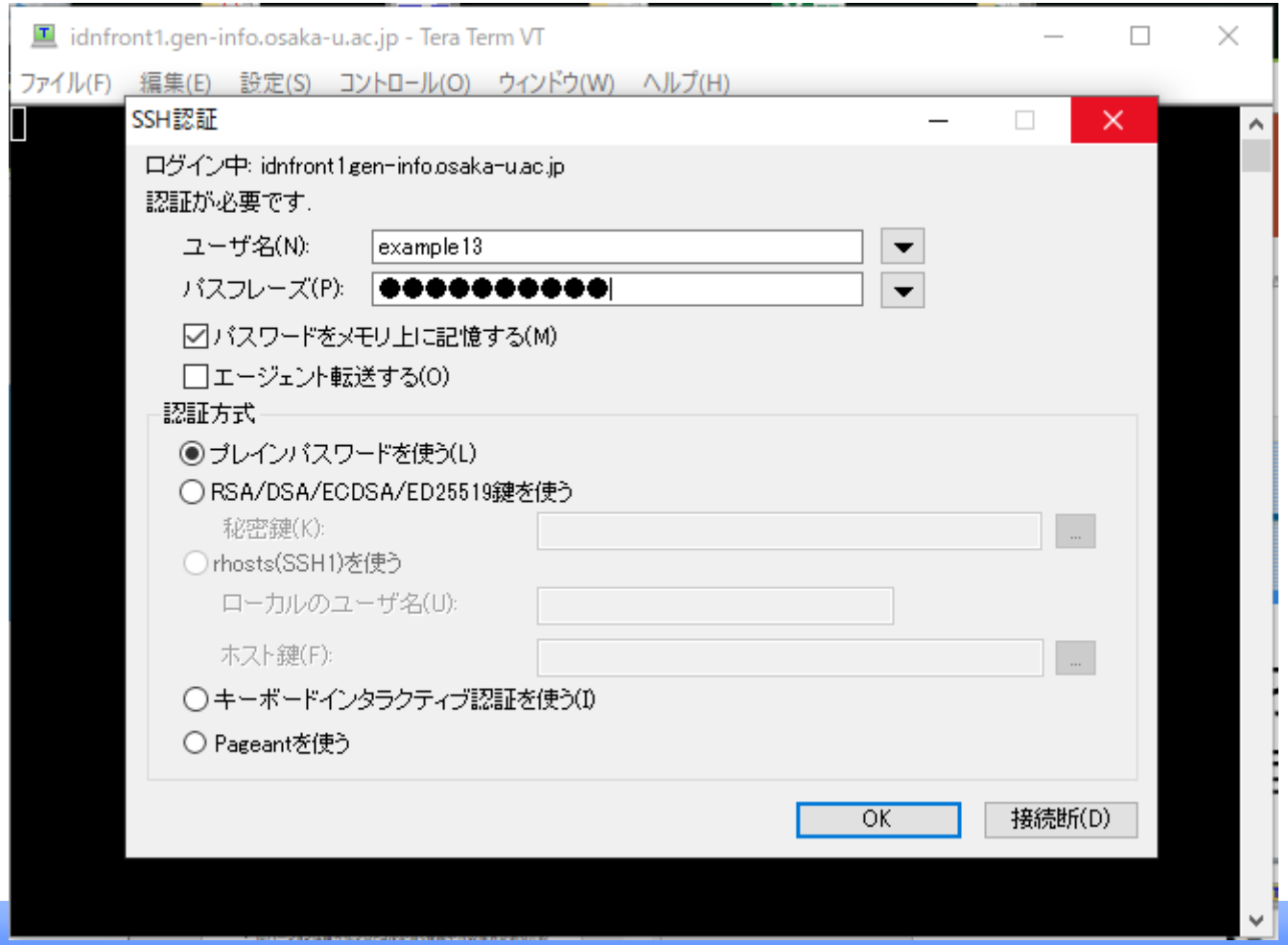
Tera Termによるサーバー接続

- 初回接続時のみ「セキュリティ警告」が出るが「続行(C)」する
  - 2回目以降は出ない



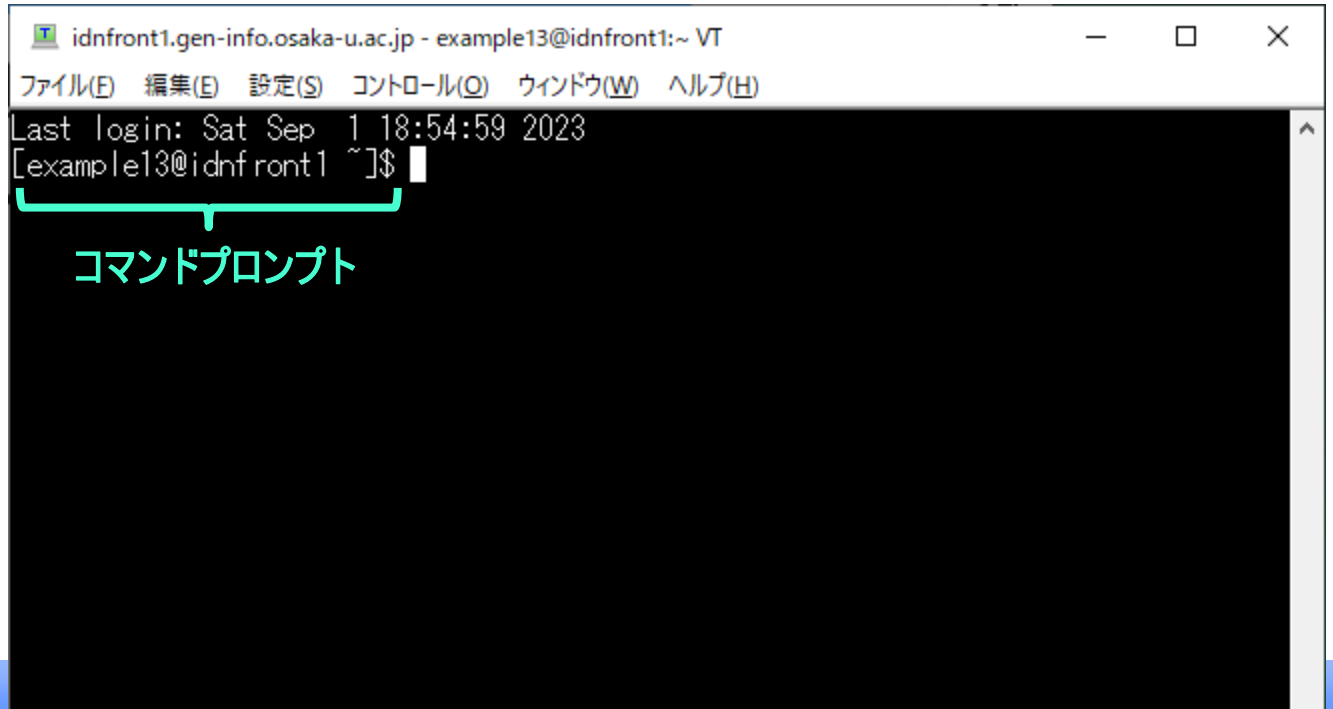
Tera Termによるサーバー接続

- 「ユーザ名(N)」に各自のIDを入力
- 「パスフレーズ(P)」にパスワードを入力
- 「認証方式」は「プレインパスワードを使う(L)」のまま



Tera Termによるコマンドライン操作

- 黒背景で白文字表示された「ターミナル」の画面が出る
- 「コマンド」を入力して計算機を操作（コマンドライン操作）
- [xxxx@idnfront1 ~]$ の部分を「コマンドプロンプト」と言う
  - コマンドの入力を受付可能である合図
  - 表示内容はカスタマイズ可能
  - デフォルトでは「~」の部分はディレクトリを移動すると表示が変化



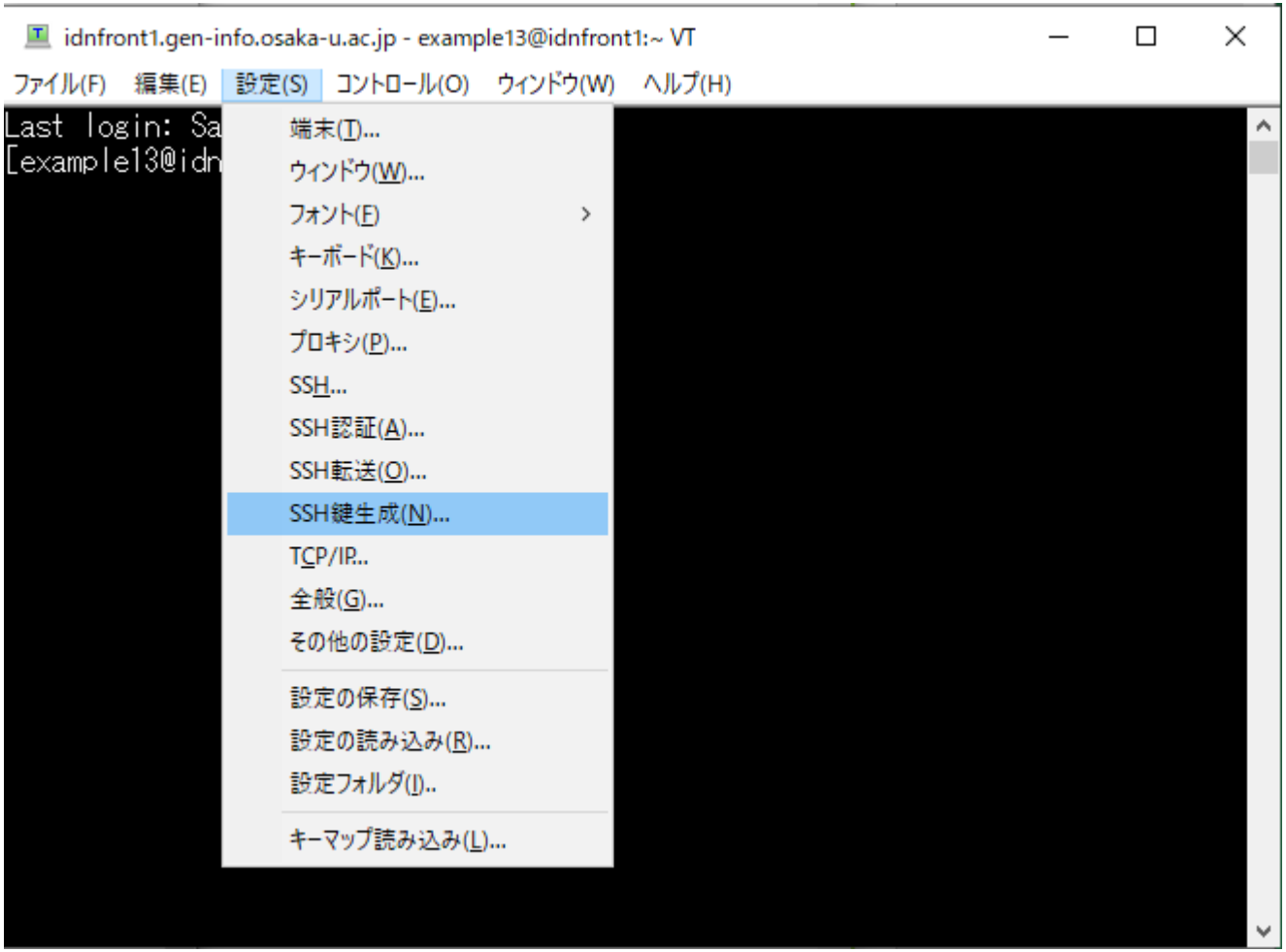
SSH鍵認証の設定

- SSHの鍵認証（公開鍵認証）とは？
  - 公開鍵暗号化方式による認証
  - 総当たり攻撃・辞書攻撃が実質不可能
    - パスワード認証は総当たり攻撃・辞書攻撃に弱い

- 計算機システムへの学外からの接続は鍵認証のみ許可
  - 学内ネットワークからはパスワード認証も可能
  - 接続元を問わずセキュリティの高い鍵認証を推奨

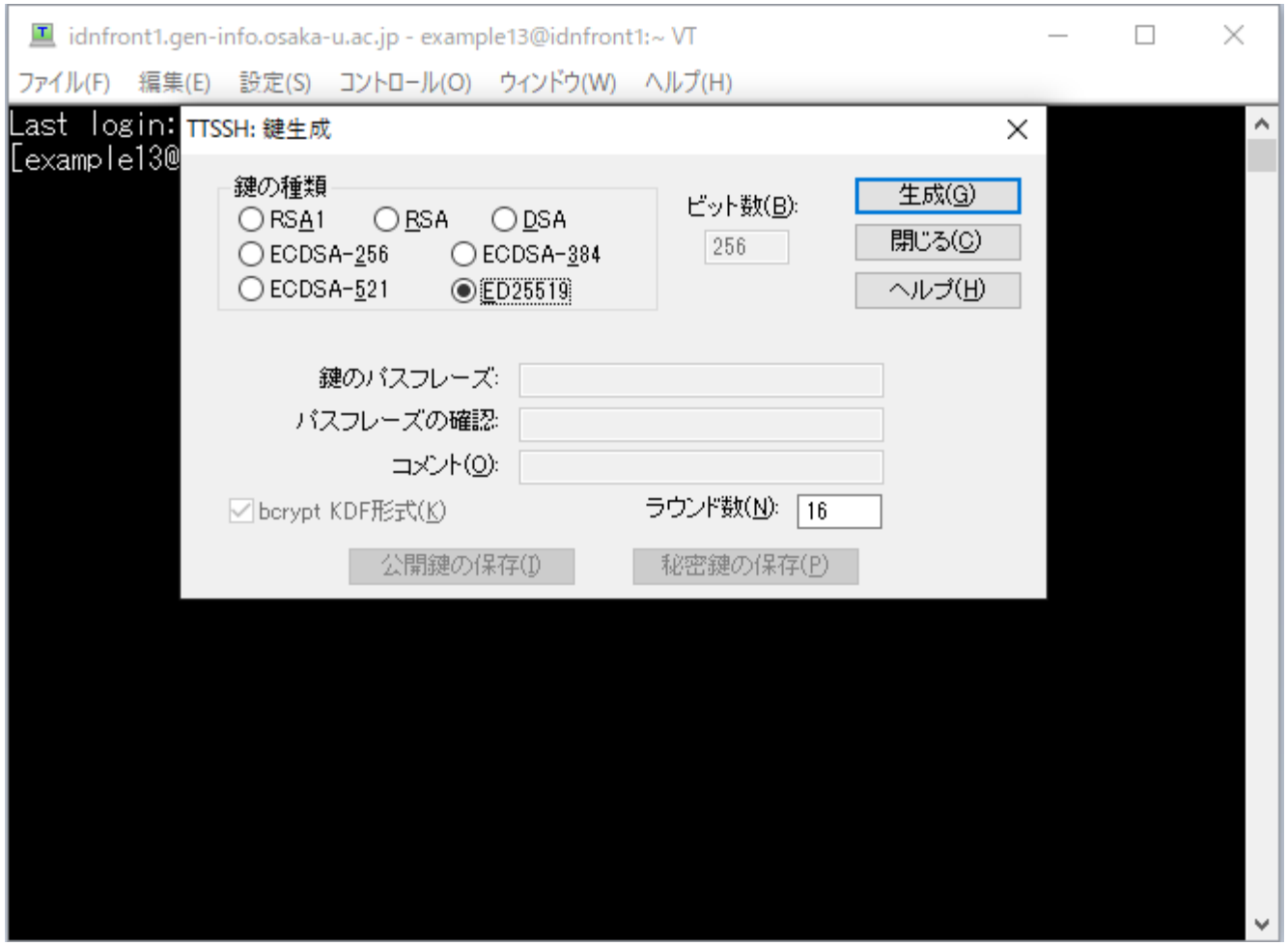
TeraTermによるSSH鍵認証の設定

- 「設定(S)」→「SSH鍵作成(N)」



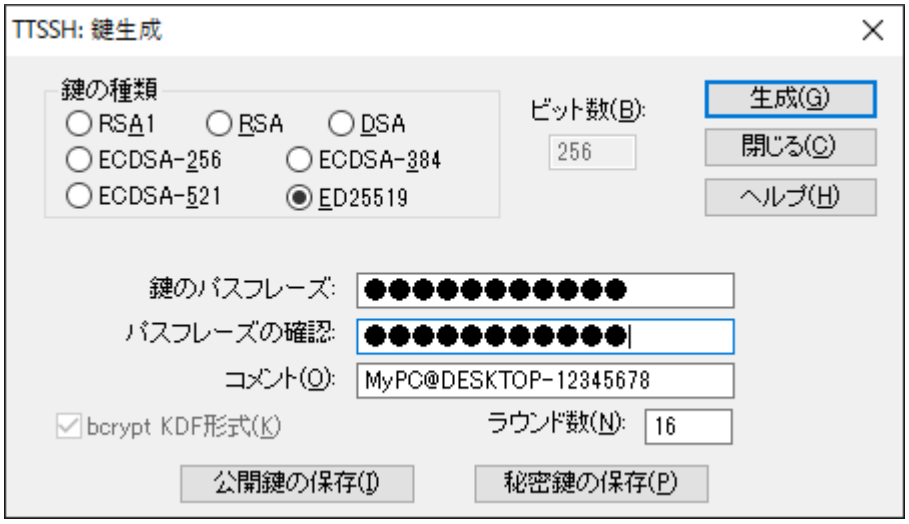
TeraTermによるSSH鍵認証の設定

- 「鍵の種類」:「ED25519」を選択
- 「生成(G)」を押す



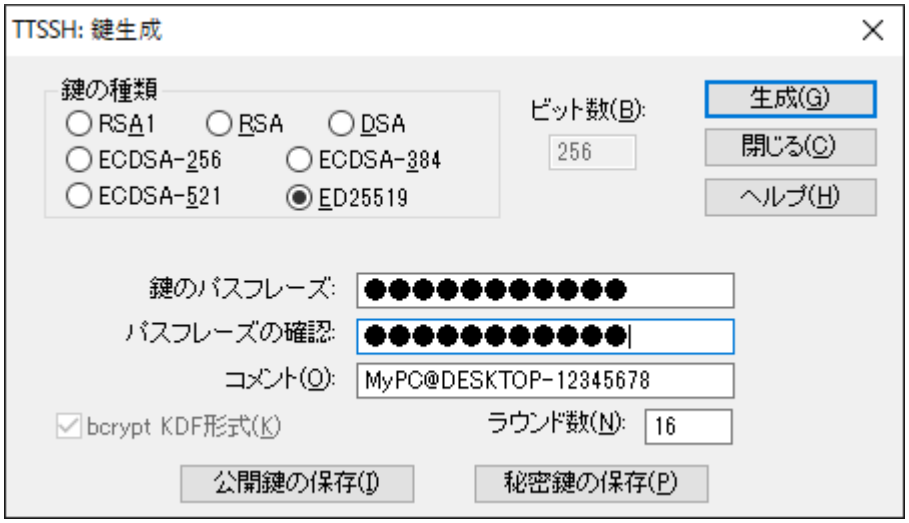
TeraTermによるSSH鍵認証の設定

- 「生成(G)」を押すと「鍵のパスフレーズ」等が入力可能となる
- 「鍵のパスフレーズ」に鍵ファイル用のパスフレーズを入力
  - 注意：計算機システムのパスワードとは異なるものを作成し入力！
  - パスフレーズはメモを取るなどして忘れないようにしてください
- 「パスフレーズの確認」に同じパスフレーズを入力
- 「コメント」は内容自由（半角英数字のみ）：パソコン名を含めるなどして、他と区別可能にするのが推奨



TeraTermによるSSH鍵認証の設定

- 「公開鍵の保存(I)」で公開鍵を保存
  - 保存先は、たとえば「ドキュメント」フォルダに「ssh」の名前でフォルダを作成し、そこに保存
  - ファイル名は既定の「id_ed25519.pub」から変更不要
- 「秘密鍵の保存(P)」で秘密鍵を保存
  - 上記と同じフォルダに保存
  - ファイル名は既定の「id_ed25519」から変更不要
- 「閉じる(C)」を押す

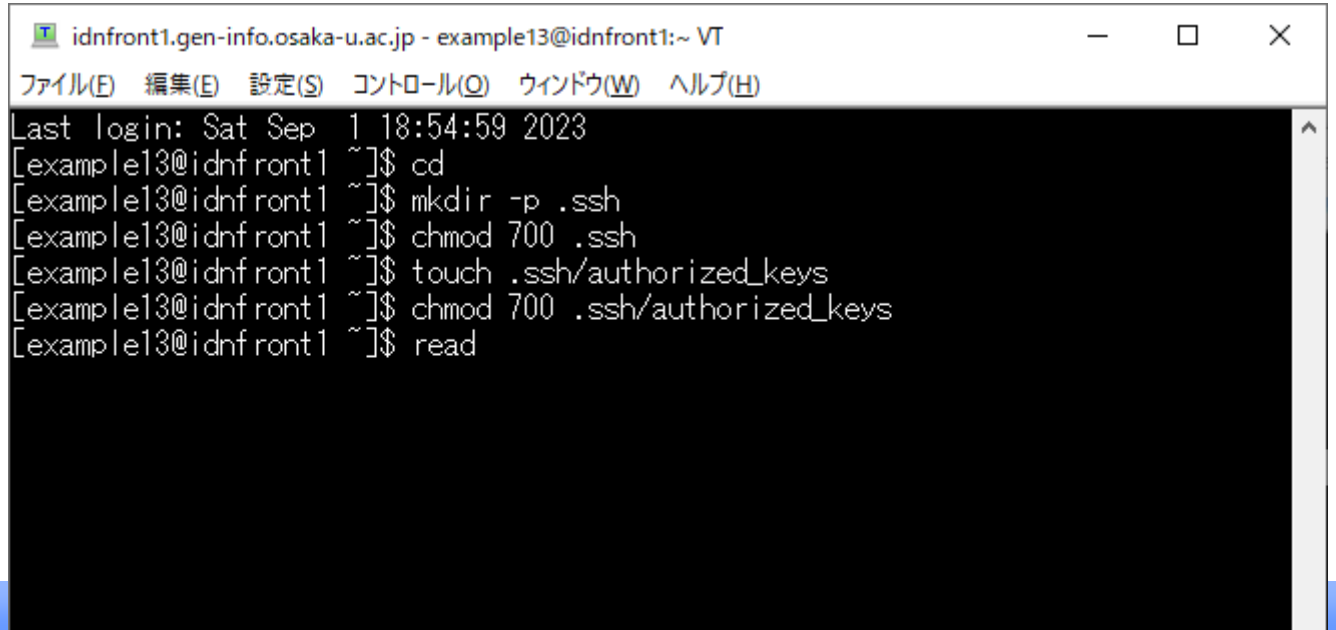


TeraTermによるSSH鍵認証の設定

- 以下のコマンドを1行ずつ入力

※ログインシェル/bin/bashの場合

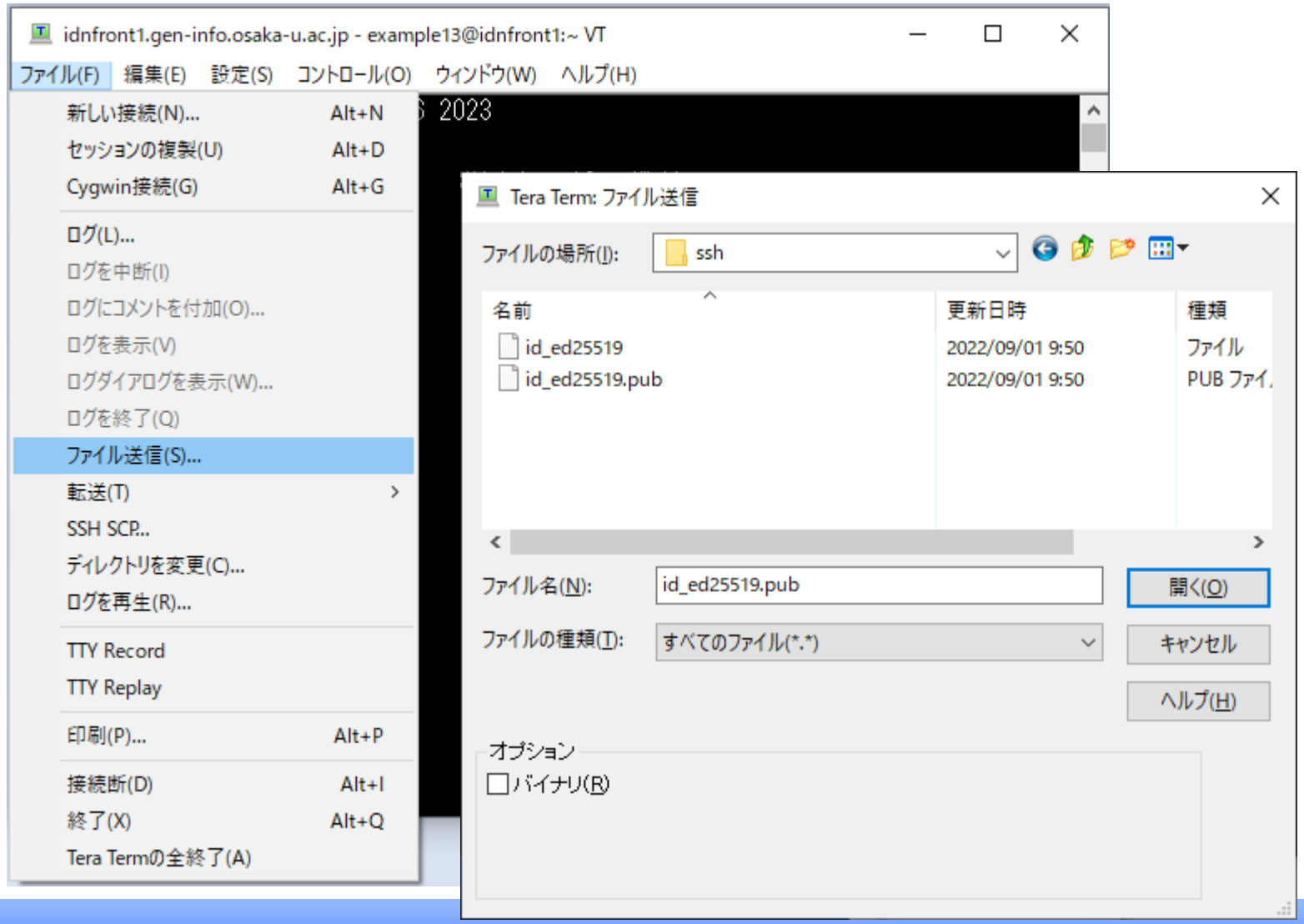
```
cd
mkdir -p .ssh
chmod 700 .ssh
touch .ssh/authorized_keys
chmod 700 .ssh/authorized_keys
read
```



※ログインシェル
/bin/cshでは
最下行のreadを
以下に変更
set REPLY=$<

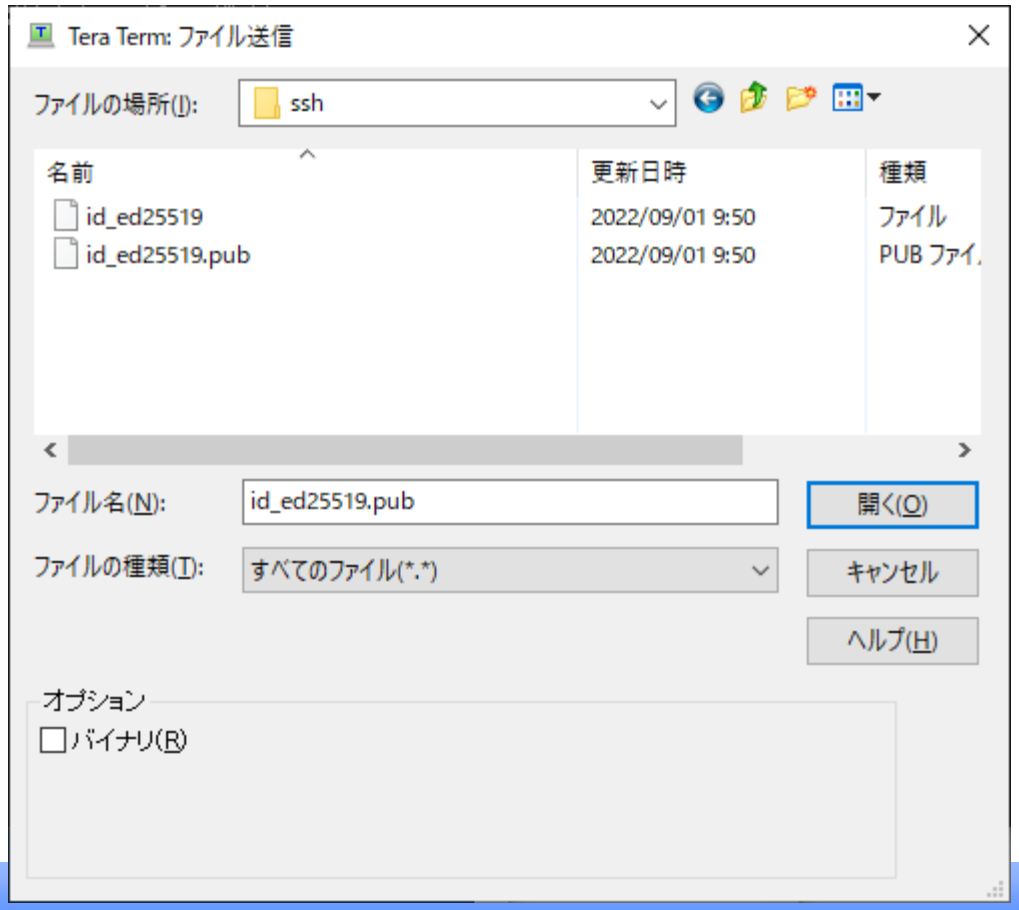
TeraTermによるSSH鍵認証の設定

- 「ファイル(F)」→「ファイル送信(S)…」



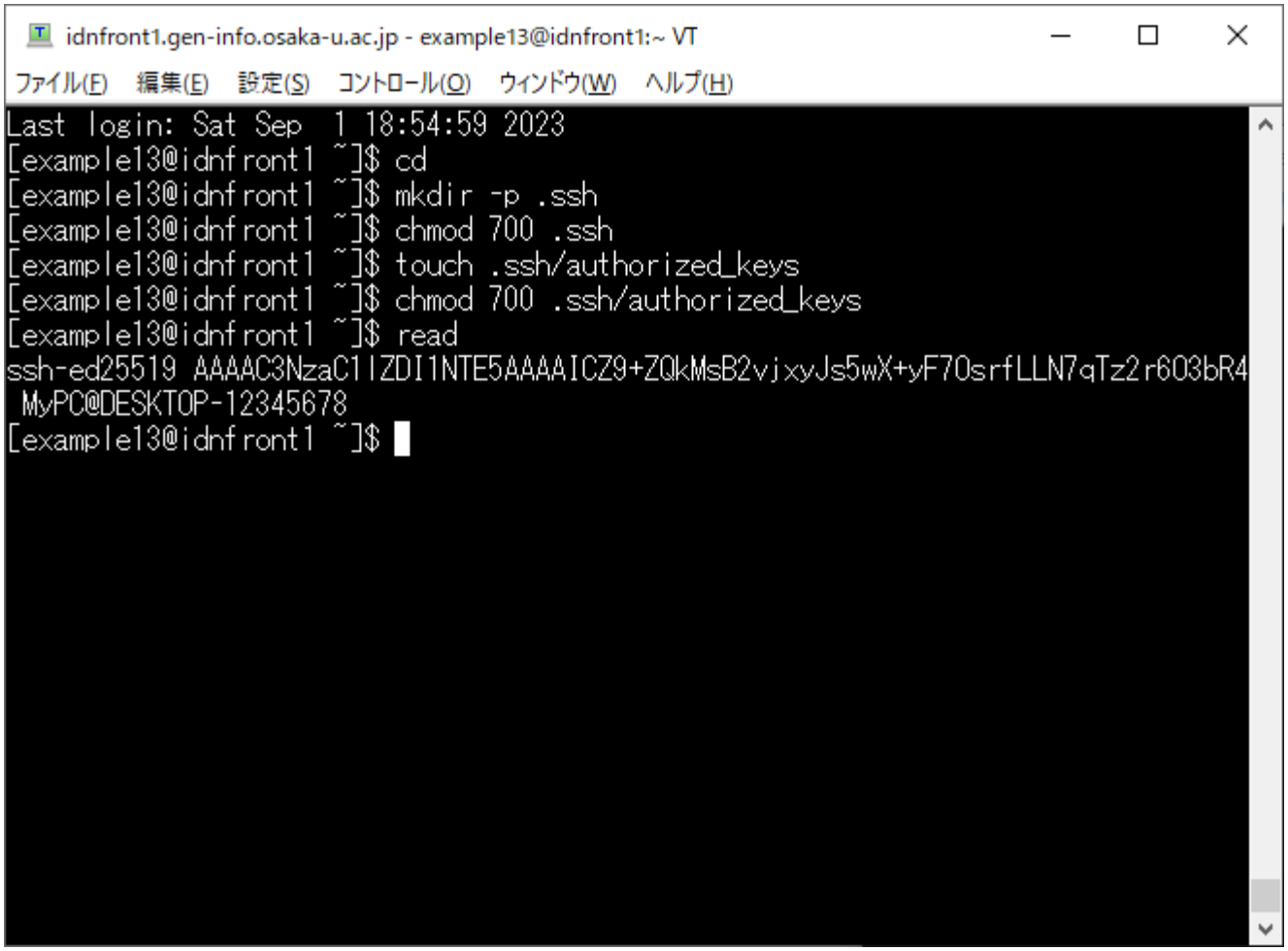
TeraTermによるSSH鍵認証の設定

- 「Tera Term ファイル送信」のウインドウが出る
- オプションの「バイナリ(R)」はチェックを外したまま
- 先ほどの保存フォルダを選び、「id_ed25519.pub」を選ぶ
- 「開く」を押す



TeraTermによるSSH鍵認証の設定

- 以下のように公開鍵の内容が1行送信されたのを確認
  - （鍵の内容は人により異なる）



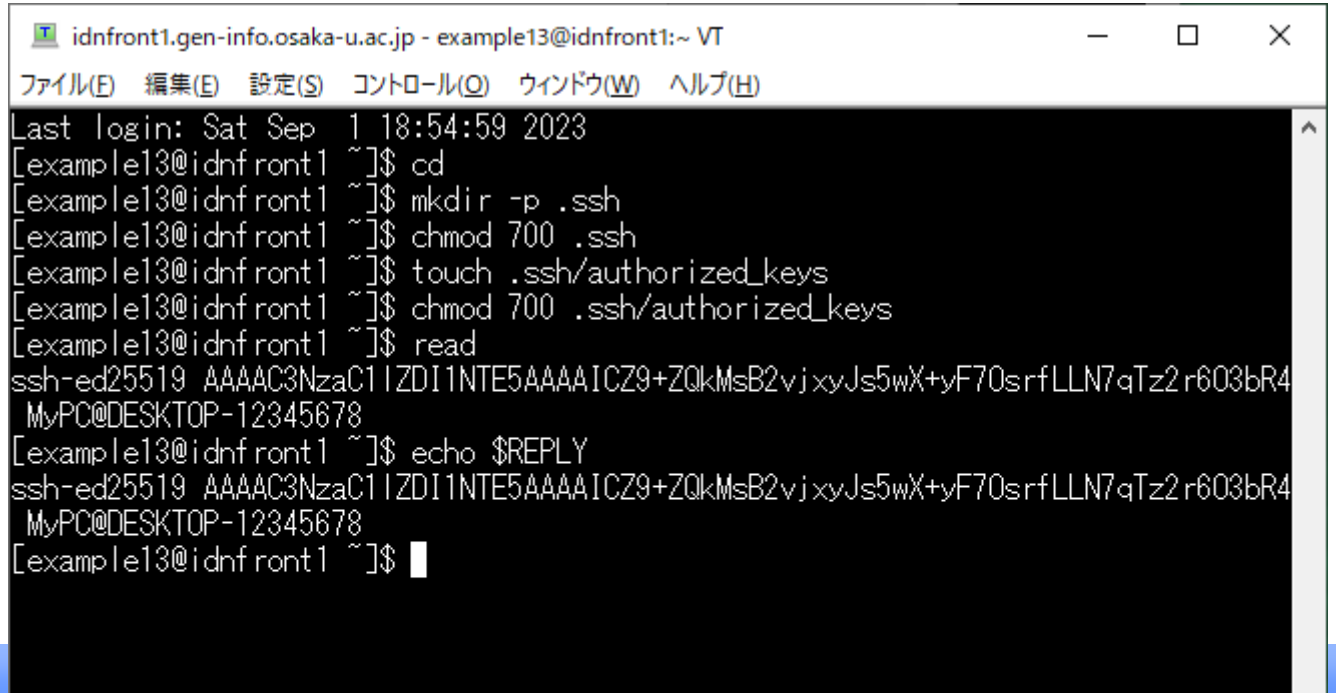
TeraTermによるSSH鍵認証の設定

- 以下のコマンドを入力し、さきほどの送信内容＝公開鍵の内容がそのまま表示される点を確認

```
echo $REPLY
```

- OKなら以下のコマンドを入力し公開鍵をファイルに書き込む

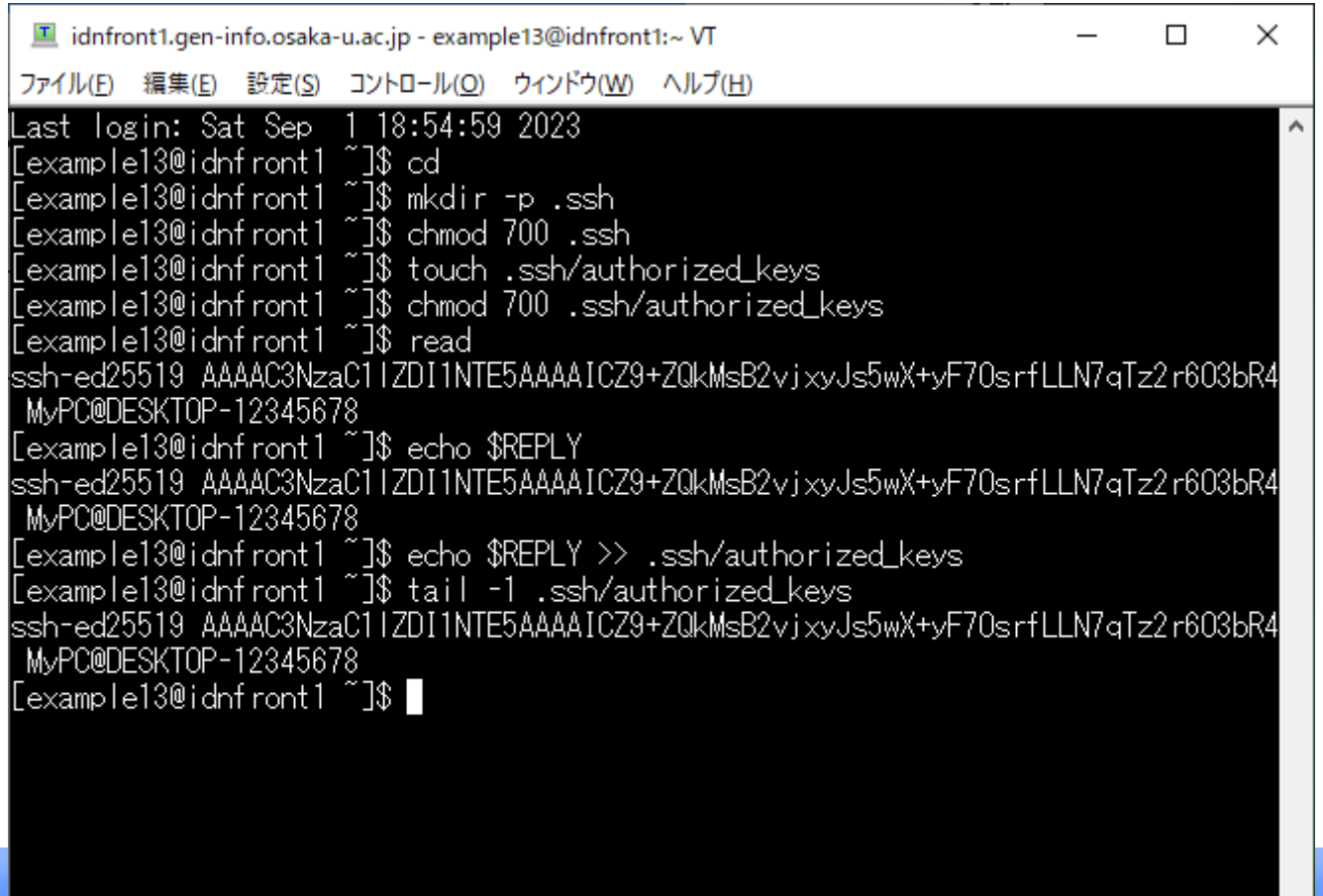
```
echo $REPLY >> .ssh/authorized_keys
```



TeraTermによるSSH鍵認証の設定

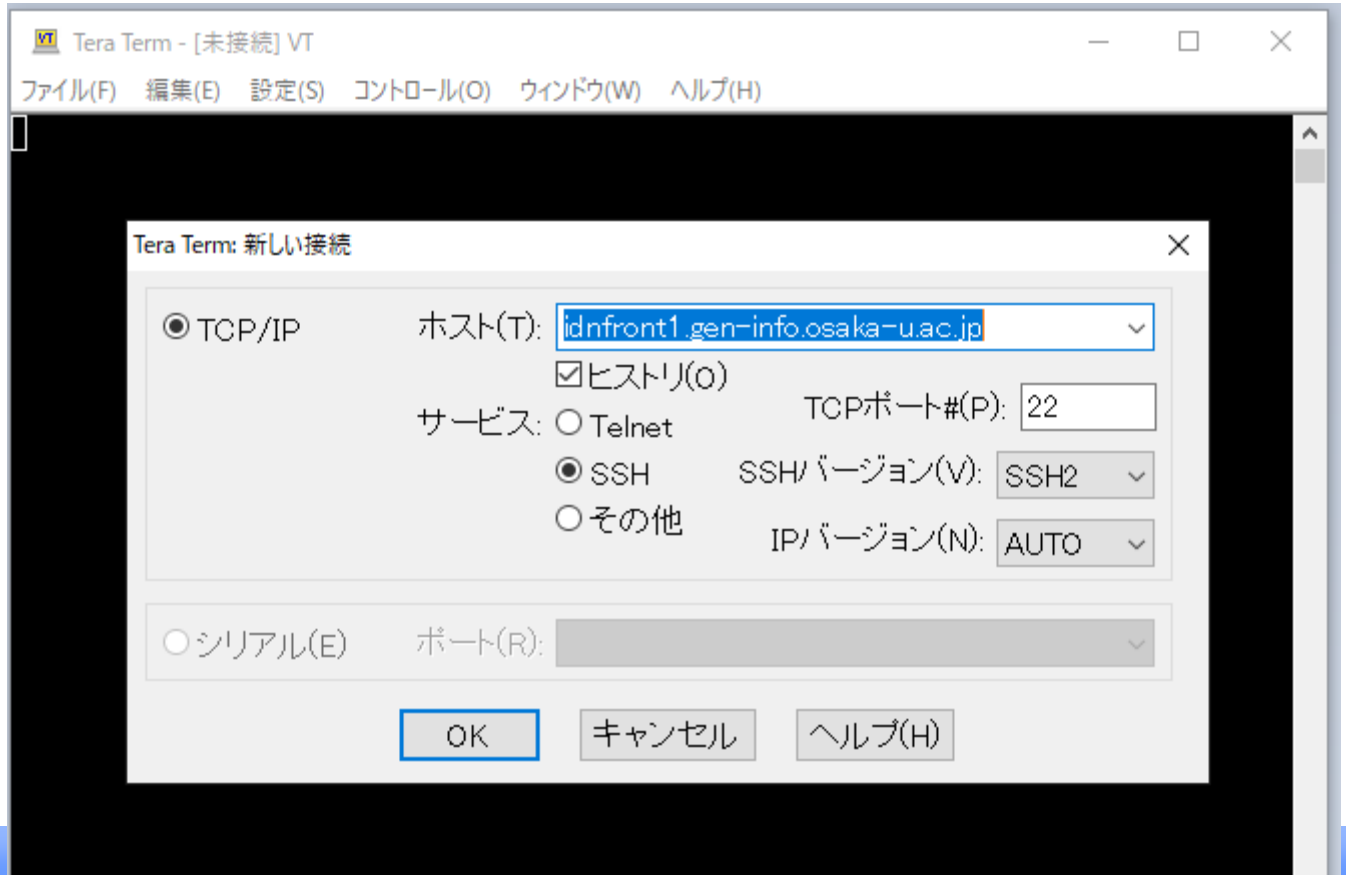
- 公開鍵の内容がファイルに書き込まれたか最終確認
  - 先ほどと同じ公開鍵の行が表示されたらOK

```
tail -1 .ssh/authorized_keys
```



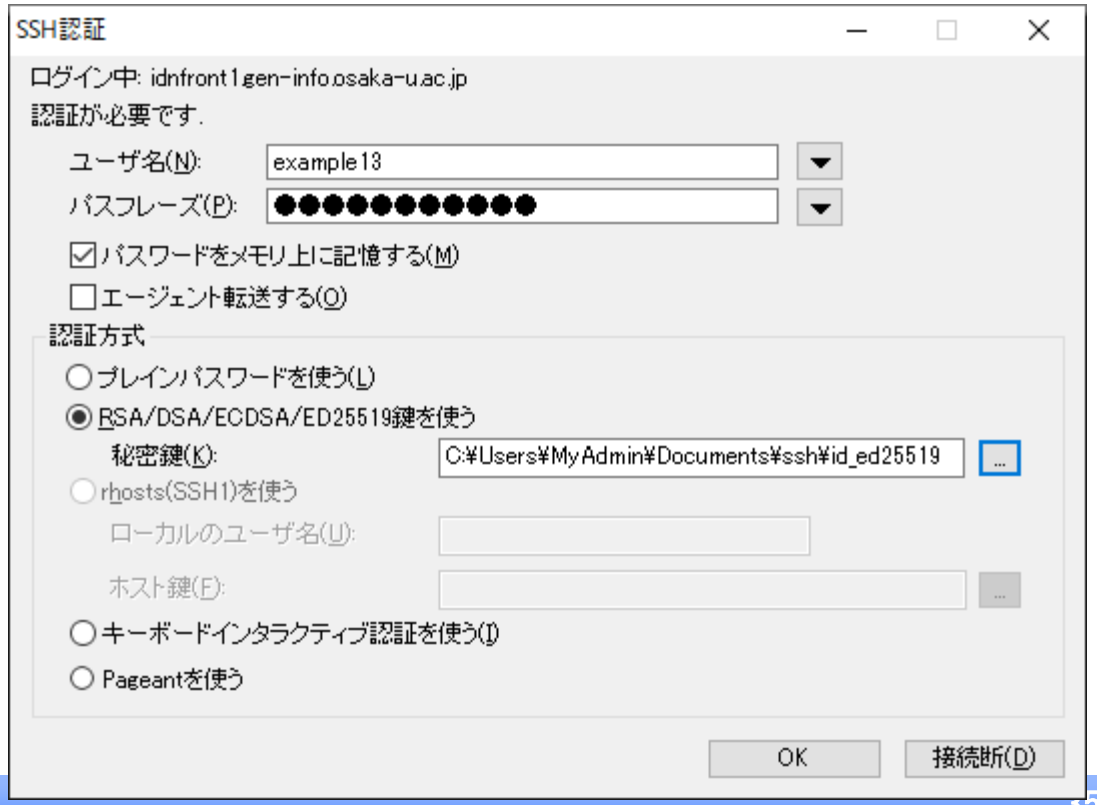
鍵認証によるログイン

- TeraTermを新規に起動
- 「Tera Term 新しい接続」のウインドウが出る
- 「ホスト(T)」に「idnfront1.gen-info.osaka-u.ac.jp」が選択されている点を確認して「OK」を押す



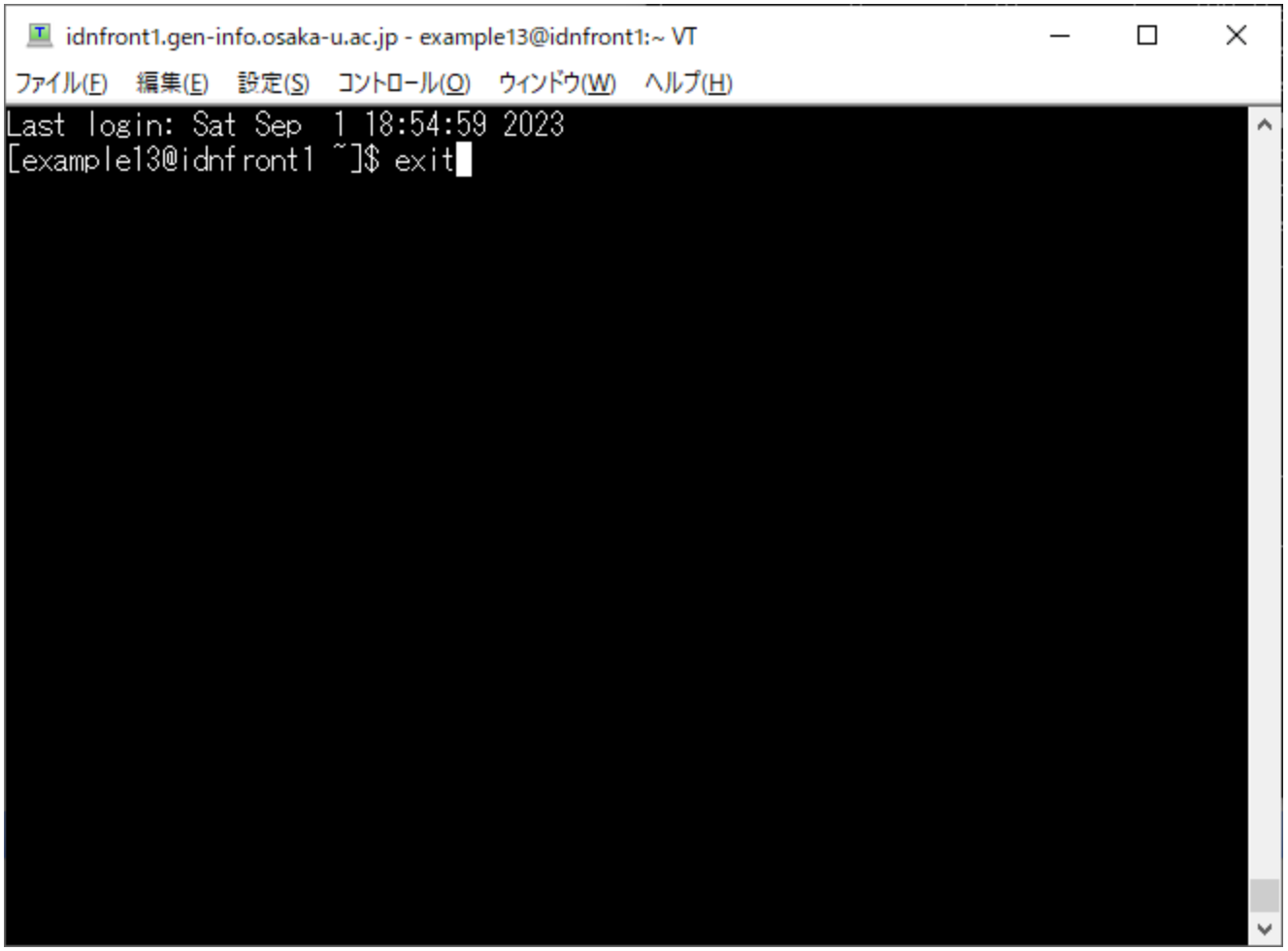
鍵認証によるログイン

- 「ユーザ名(N)」:ID
- 「パスフレーズ(P)」:鍵のパスフレーズ
- 「認証方式」:「RSA/DSA/ECDSA/ED25519鍵を使う」を選択
- 「 秘密鍵(K)」の右端の「…」を押して、鍵の保存先フォルダのファイル「id_ed25519」を選択して「開く」
- 「OK」を押す
- ログイン出来たら成功



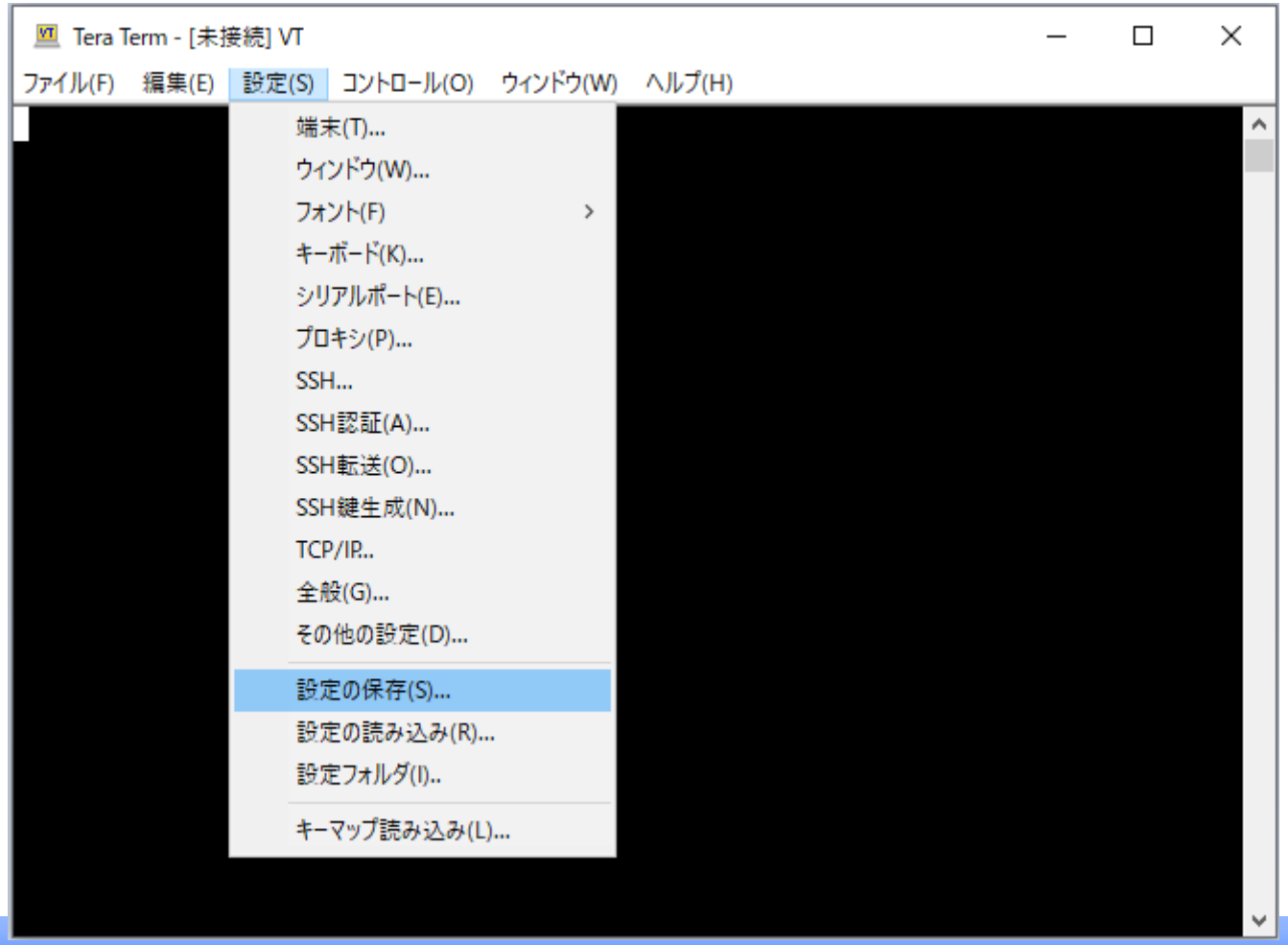
接続の切断方法

- 「exit」と入力してEnterキーを押す
- ウインドウ右上の「×」ボタンで強制切断しても構いません



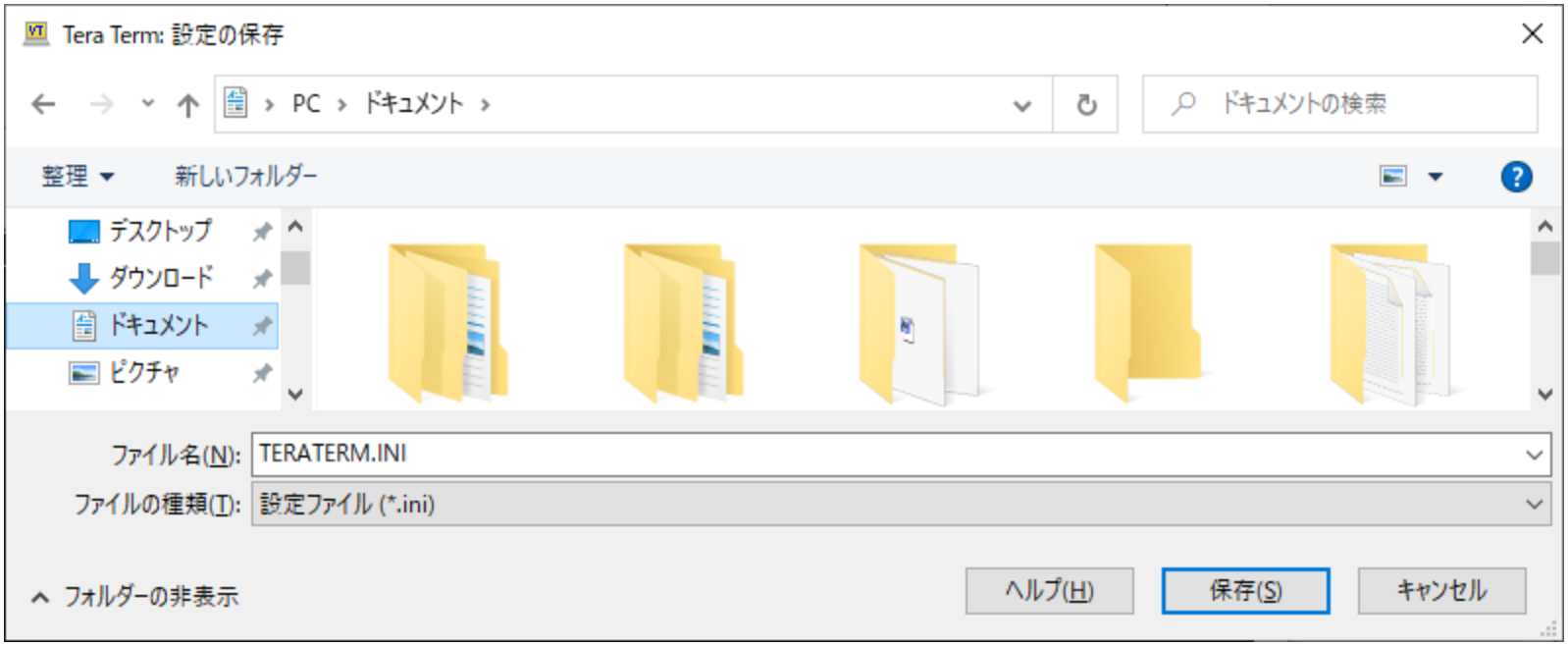
TeraTerm設定の保存

- 鍵ファイル選択やID入力を毎回行うのは面倒のため設定を保存する
- TeraTerm起動後、「新しい接続」のウインドウが出たら「キャンセル」
- 「設定(S)」→「設定の保存」



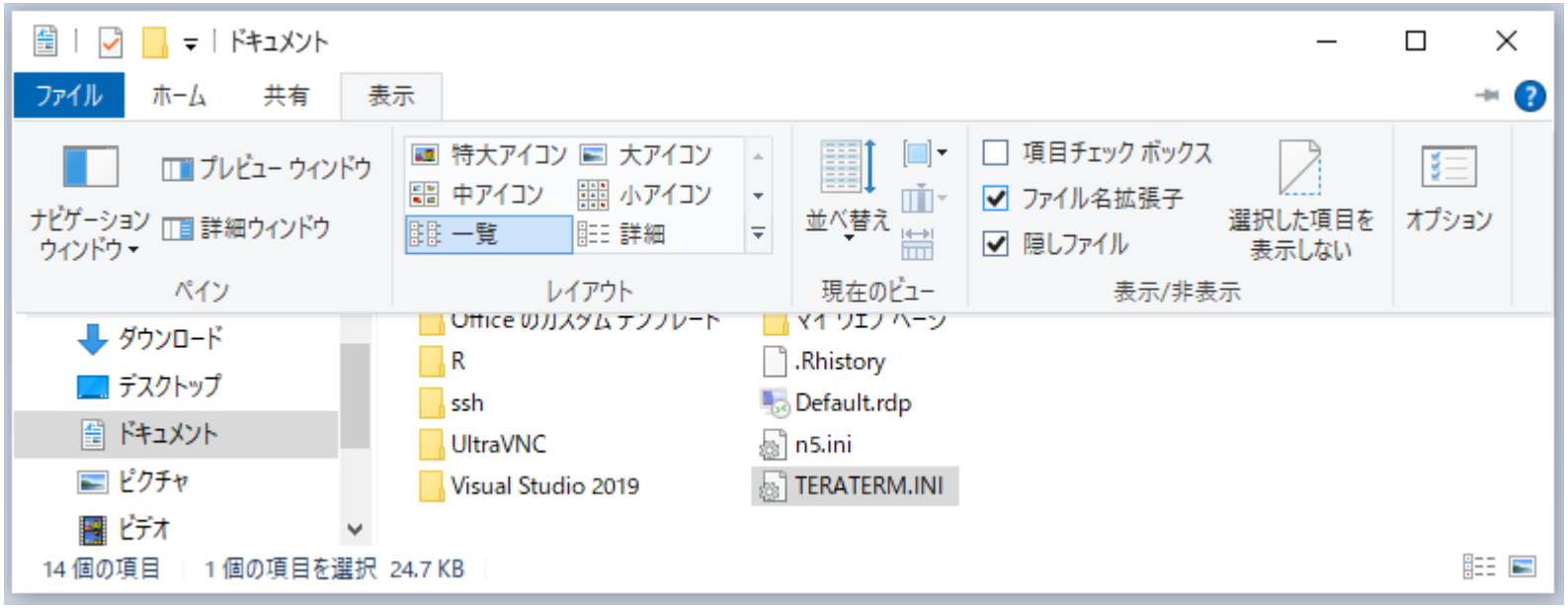
TeraTerm設定の保存

- 保存先としてWindows既定の「ドキュメント」フォルダを選択
  （TeraTermの仕様により、保存先は「ドキュメント」フォルダに限定となります）
- ファイル名はデフォルトの「TERATERM.INI」のまま変更不要
- 「保存(S)」をクリック



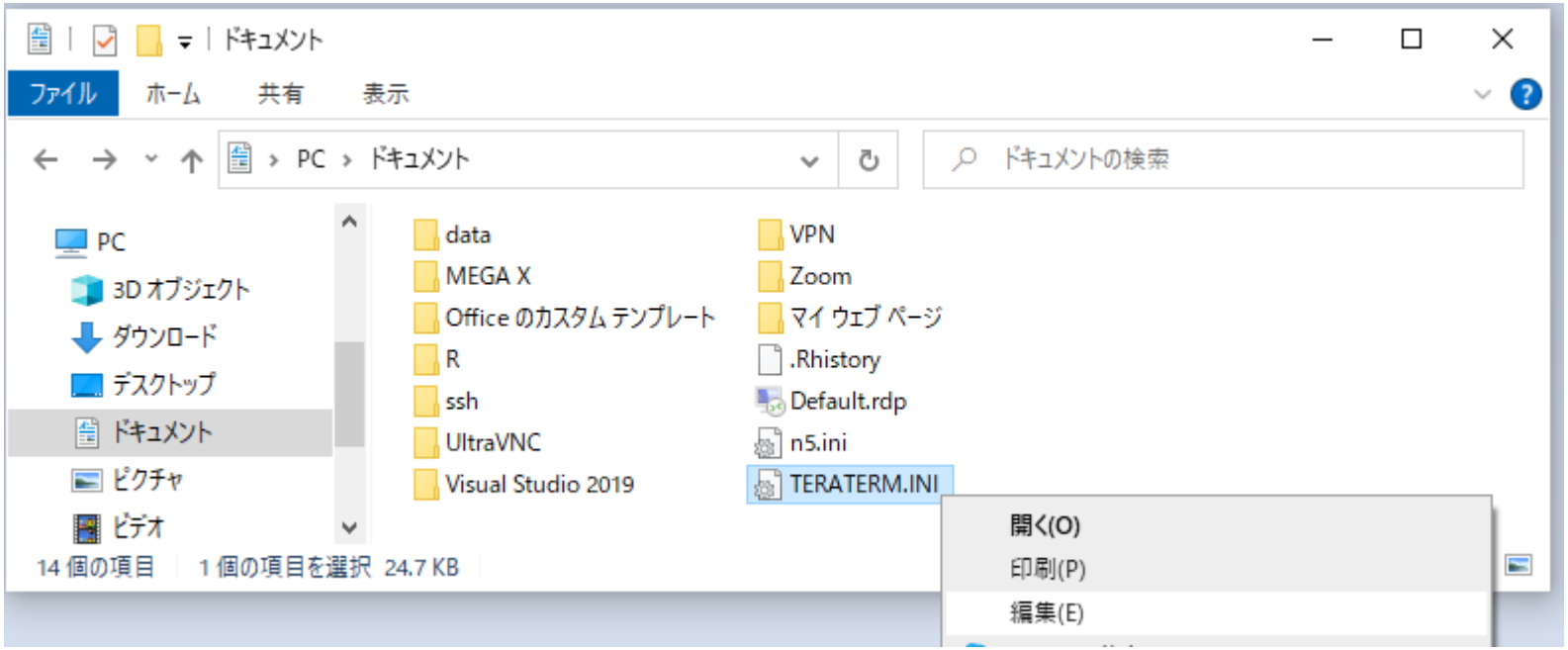
TeraTerm設定の保存

- エクスプローラで「ドキュメント」を開く
- 「表示」の「ファイル名拡張子」のチェックを入れる



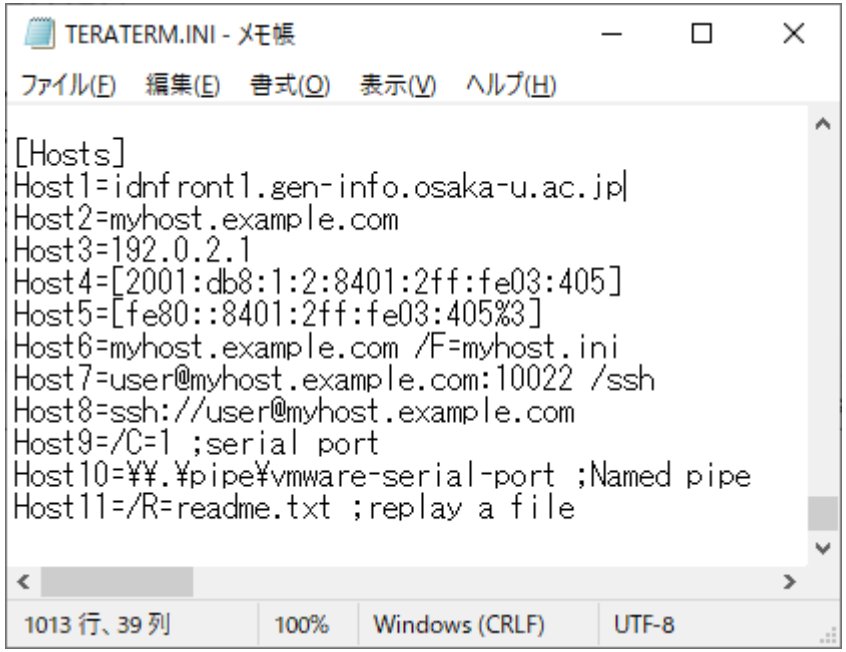
TeraTerm設定の保存

- TERATERM.INI を右クリックして「編集(E)」をクリック



TeraTerm設定の保存

- TERATERM.INIを開いたメモ帳が起動する
- ファイル末尾にスクロールして[Hosts]の記載を探す

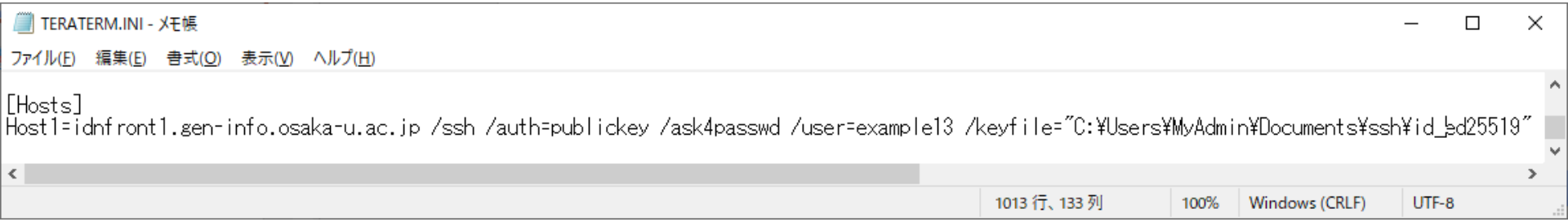


TeraTerm設定の保存

- TERATERM.INI内の[Hosts]部分を以下のように編集
  - 不要なHost2以降をすべて削除
  - Host1を以下のように変更
    - 「自分のID」は各自のID（半角英数字）を入力
    - 「秘密鍵ファイルの絶対パス」は、次ページで説明する「パスのコピー(A)」を行った直後に、メモ帳のメニューから「編集(E)」→「貼り付け(P)」でペースト

Host1=idnfront1.gen-info.osaka-u.ac.jp /ssh /auth=publickey /ask4passwd /user=自分のID /keyfile=秘密鍵ファイルの絶対パス

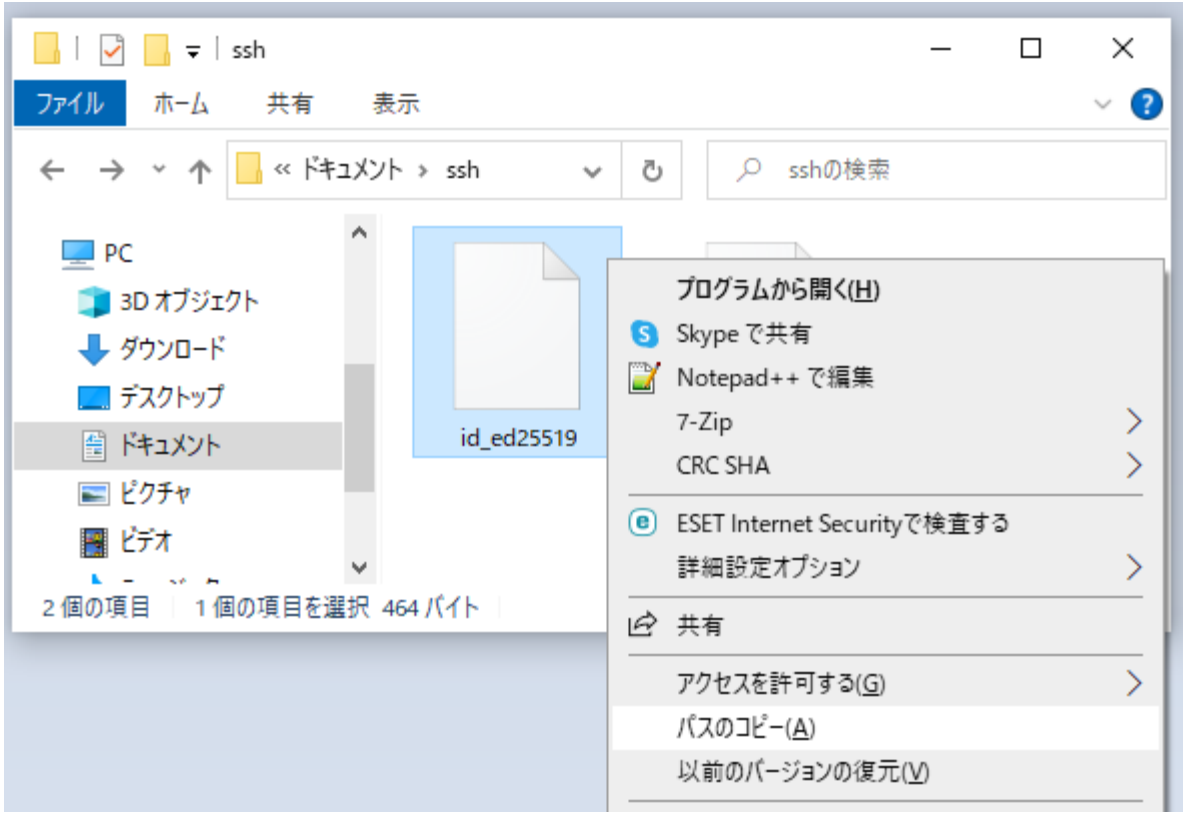
（注：表示の都合で折り返していますが、実際には1行で入力）



- パスフレーズは毎回入力が必要
  - 入力不要にする方法は存在しますが今回は触れません

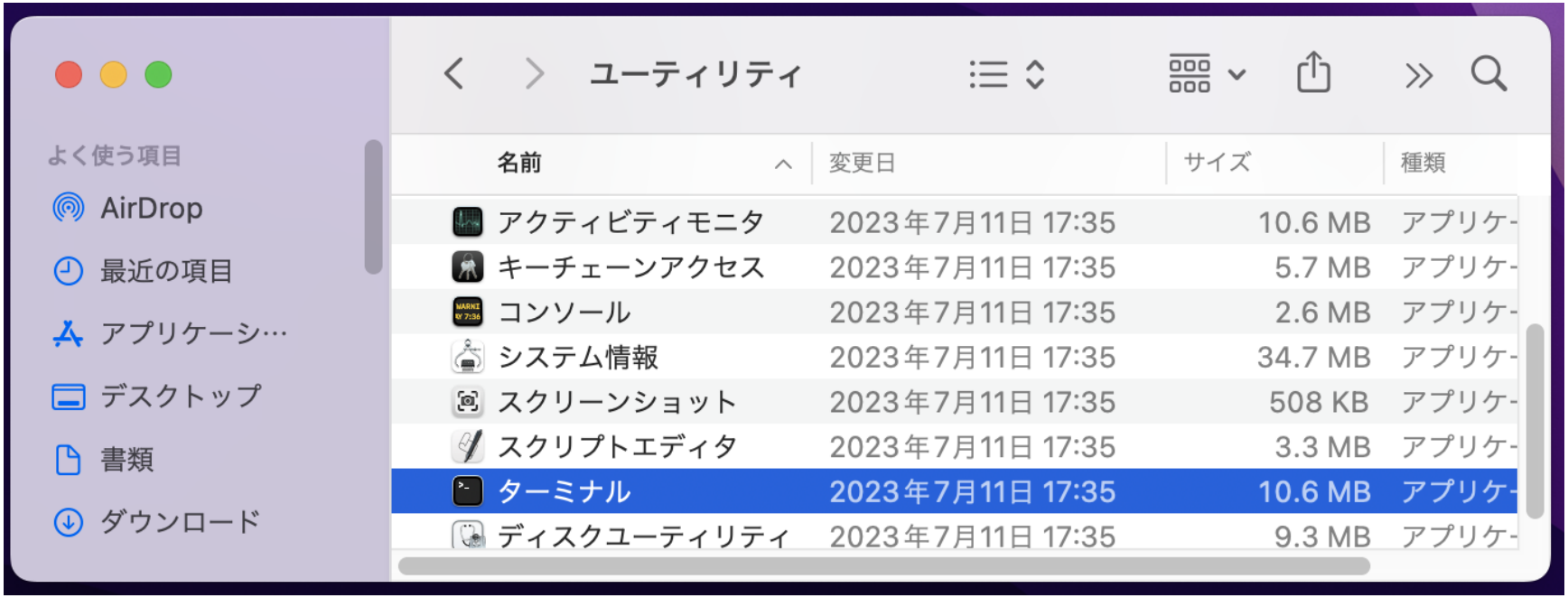
TeraTerm設定の保存

- 秘密鍵ファイルの絶対パス（場所を表す文字列）の確認
  - エクスプローラで「ドキュメント」→「ssh」（鍵ファイル保存先）を開く
  - 「id_ed25519」（拡張子無し）をShiftキーを押しながら右クリック
  - 「パスのコピー(A)」をクリック



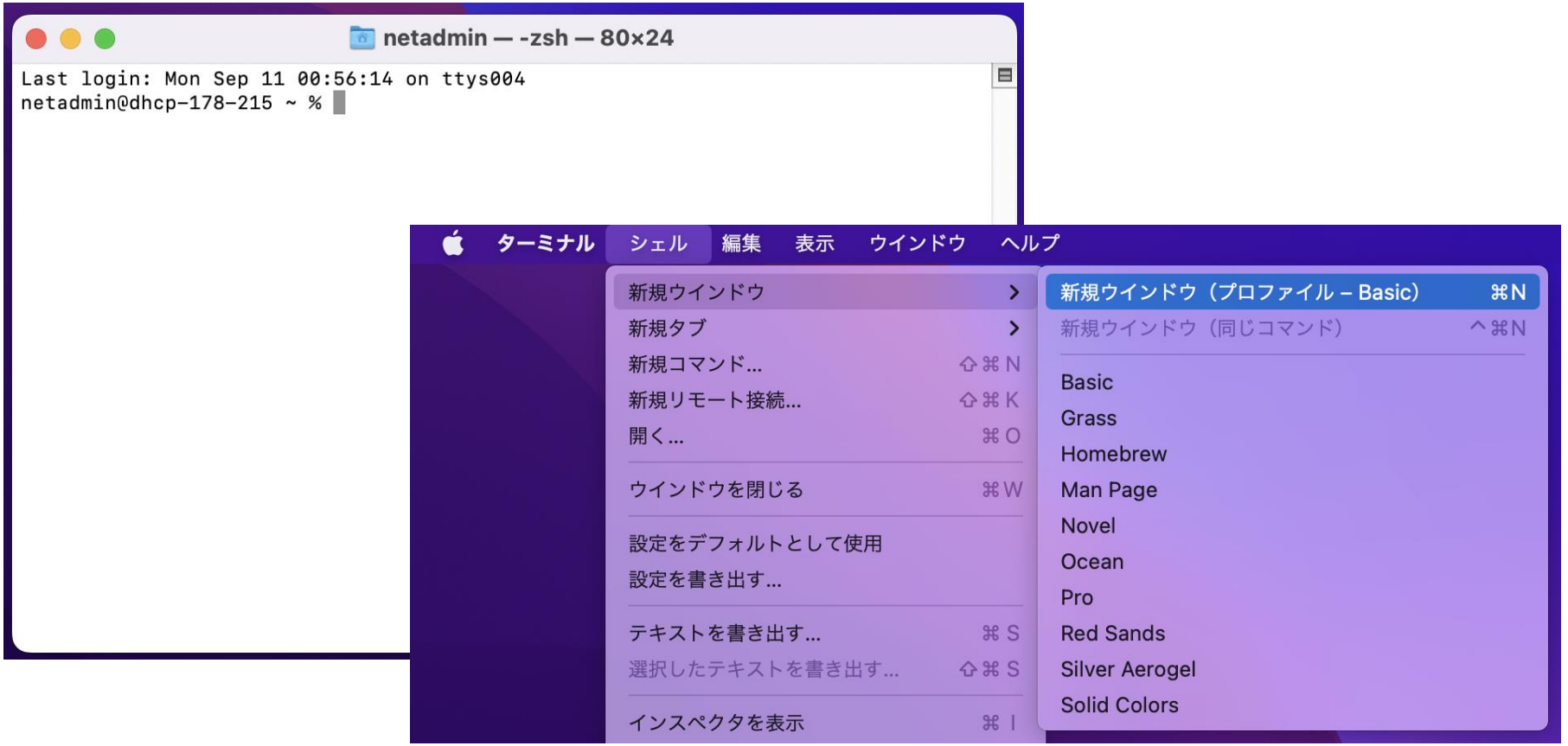
MacのターミナルによるSSH接続

- Macの「ターミナル」（Terminal）は標準インストール済
  - Finderの「ファイル」→「新規Finderウインドウ」を開いて、「アプリケーション」→「ユーティリティ」に「ターミナル」がある
  - ドックにドラッグ＆ドロップして追加するのを推奨
- 元々はMacパソコンのコマンドライン操作用途
  - Mac標準搭載のsshコマンドと組み合わせてリモートSSH接続を実現



MacのターミナルによるSSH接続

- ターミナル起動直後には1個、新規ウインドウが開く
- 「シェル」→「新規ウインドウ」→「新規ウインドウ（プロファイル – Basic）」（コマンドキー+N）から何個でもウインドウを作成可能
- Macをコマンドライン操作するためのウインドウ

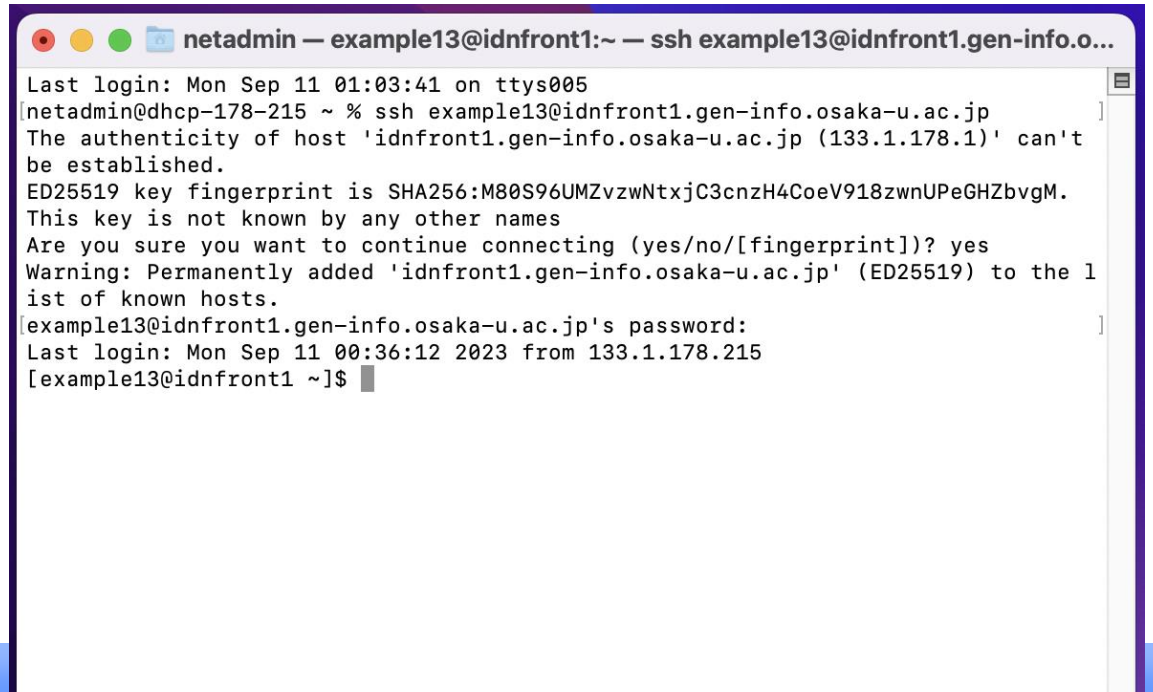


MacのターミナルによるSSH接続

- 以下のコマンドをMacのターミナルに入力
  （「自分のID」の部分はご自身のIDに置き換えてください）

```
ssh 自分のID@idnfront1.gen-info.osaka-u.ac.jp
```

  - 初回ログイン時のみ続行してよいか聞かれるので「yes」と入力
  - パスワードを尋ねられるので入力
- 終了時は「exit」を入力

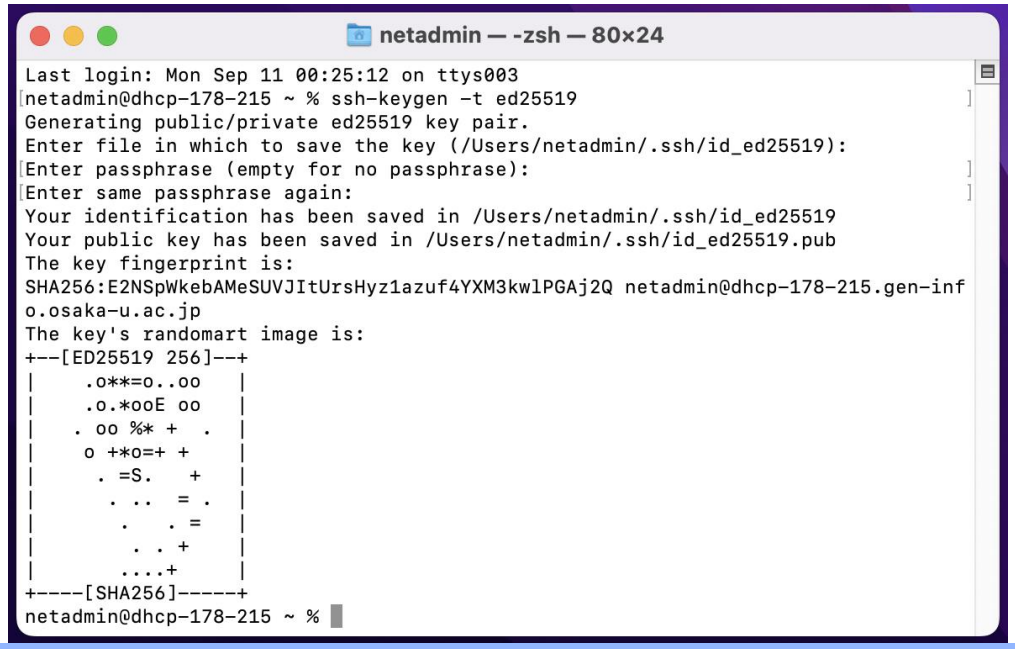


Macのターミナルによる SSH鍵の作成

- ターミナルの「シェル」→「新規ウインドウ」→「新規ウインドウ（プロファイル – Basic）」（またはコマンドキー+N）で新規ウインドウを開く
- 以下のコマンドを入力

```
ssh-keygen -t ed25519
```

- パスフレーズを聞かれるので入力
  - 注意：計算機システムのパスワードとは異なるものを作成し入力！
  - パスフレーズはメモを取るなどして忘れないようにしてください

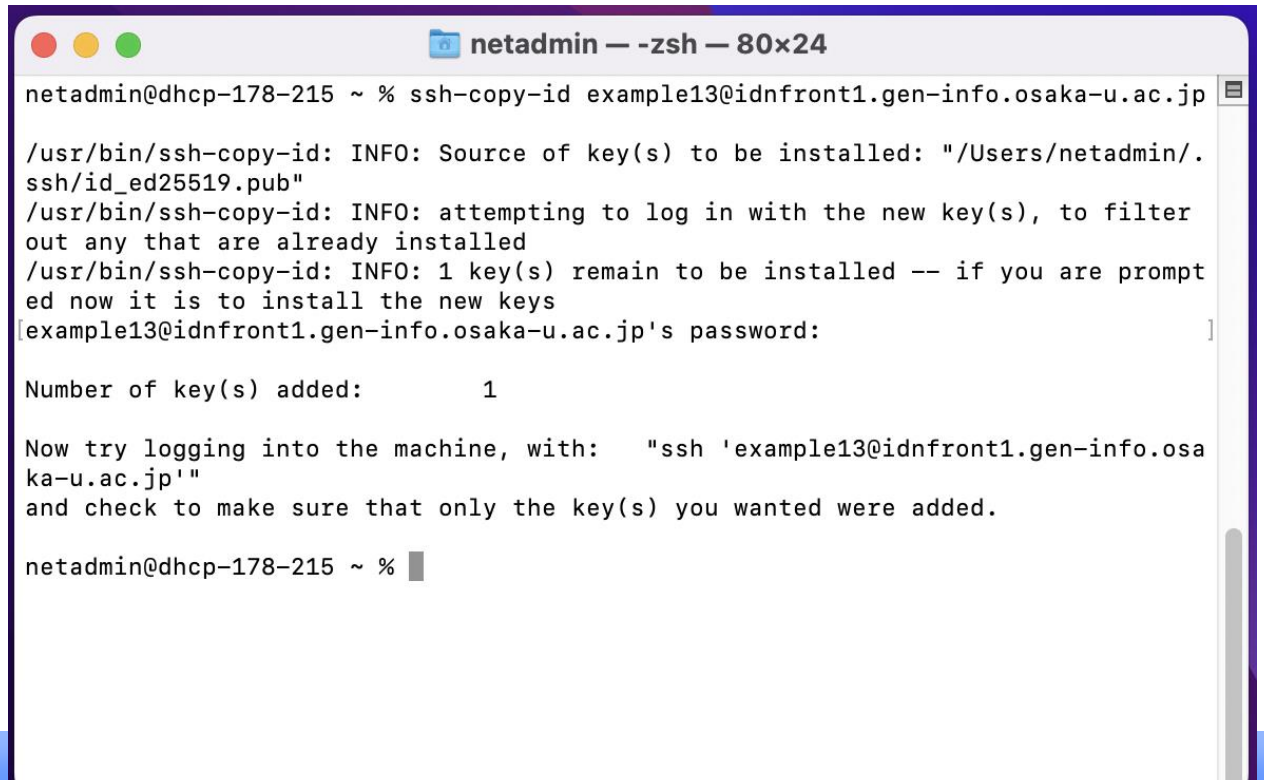


MacのターミナルによるSSH鍵のサーバー設定

- 以下のコマンドにより、作成したSSH鍵の公開鍵をサーバーに転送
（「自分のID」の部分はご自身のIDに置き換えてください）

```
ssh-copy-id 自分のID@idnfront1.gen-info.osaka-u.ac.jp
```

- パスワード（注：鍵のパスフレーズではなく、サーバーのパスワード）を尋ねられるので入力する



MacのターミナルからSSHの鍵認証による接続

- ターミナルの「シェル」→「新規ウインドウ」→「新規ウインドウ（プロファイル – Basic）」（またはコマンドキー+N）で新規ウインドウを開く
- 以下のコマンドを入力
  （「自分のID」の部分はご自身のIDに置き換えてください）

ssh 自分のID@idnfront1.gen-info.osaka-u.ac.jp

- 鍵のパスフレーズを尋ねられるので入力する
- 終了時は「exit」を入力

ファイル転送ソフトのインストール

- 「SFTP対応ファイル転送ソフト」という種類のソフト
  - 無償ソフトやフリーソフトウェアも多く存在

- Windows / Mac 両対応
  - FileZilla　https://filezilla-project.org/
  - CyberDuck　https://cyberduck.io/

- Windows
  - WinSCP　https://winscp.net/

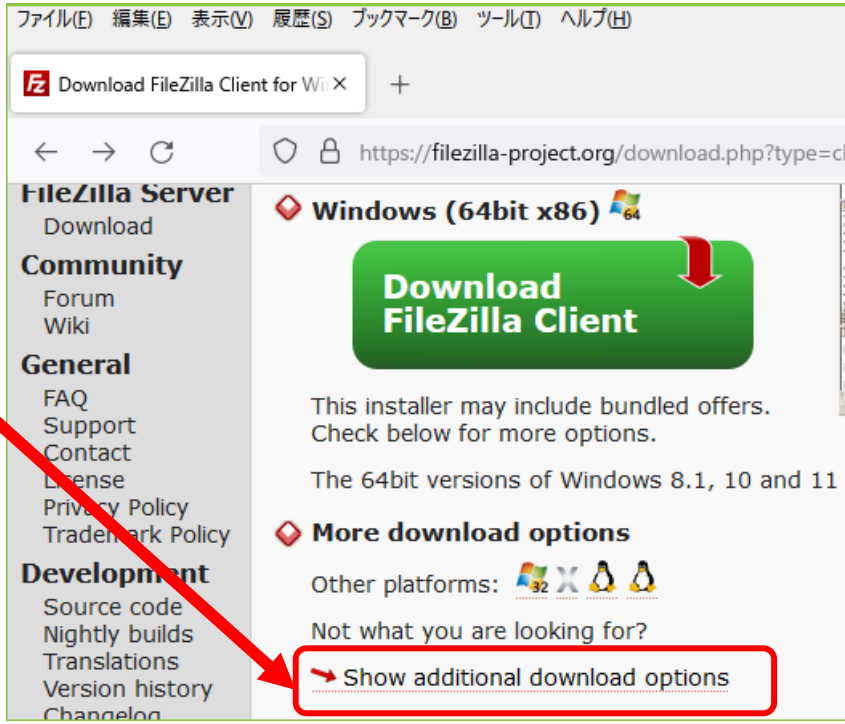
FileZilla (FileZilla Client)

- https://filezilla-project.org/
  - 様々なプロトコルに対応したファイル転送ソフト
  - Windows, Mac, Linux対応
  - フリーソフトウェア（有償のPro版もあるが通常不要）

- ダウンロード時の注意点

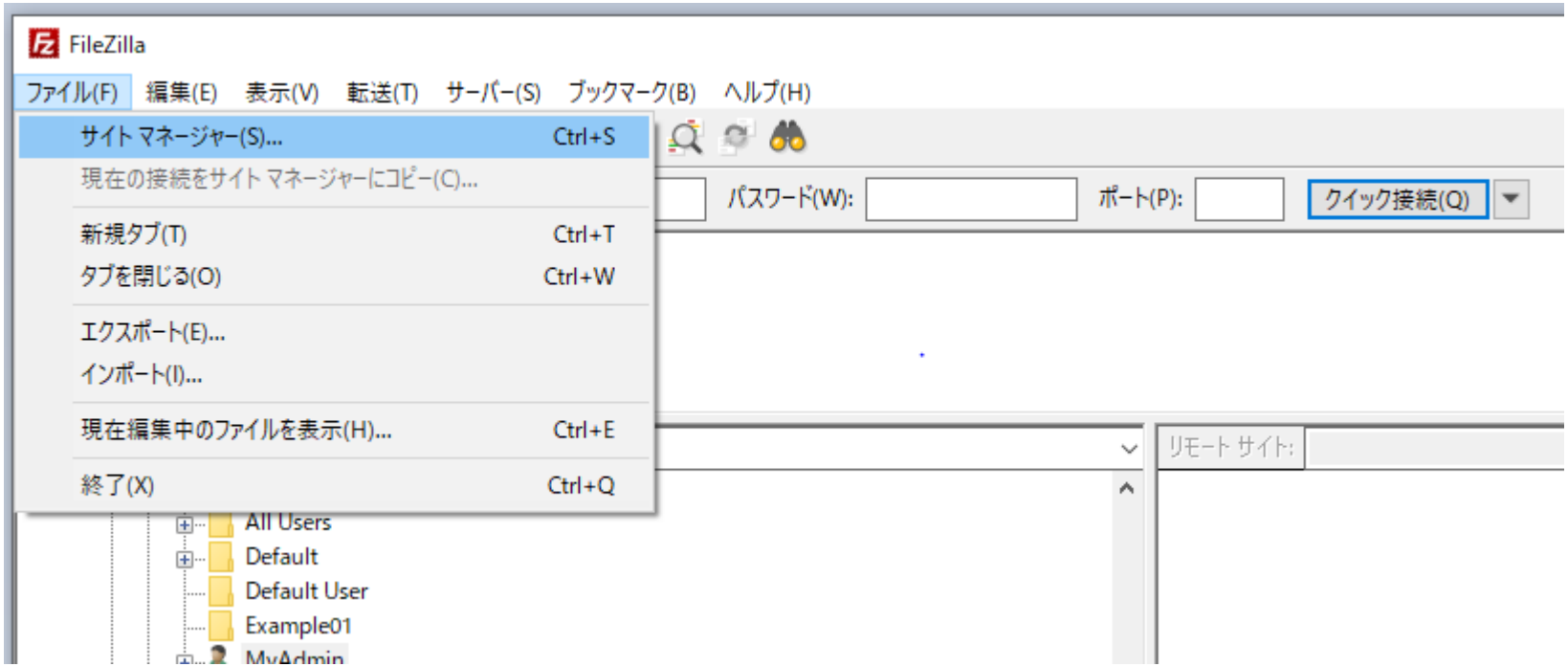
  Downloadボタンを押してダウンロードできるインストーラにはアドウェア（広告付きソフト）が含まれる。

  Show additional download options クリック後に表示されるファイルのインストーラには含まれないため、そちらのダウンロードが推奨。



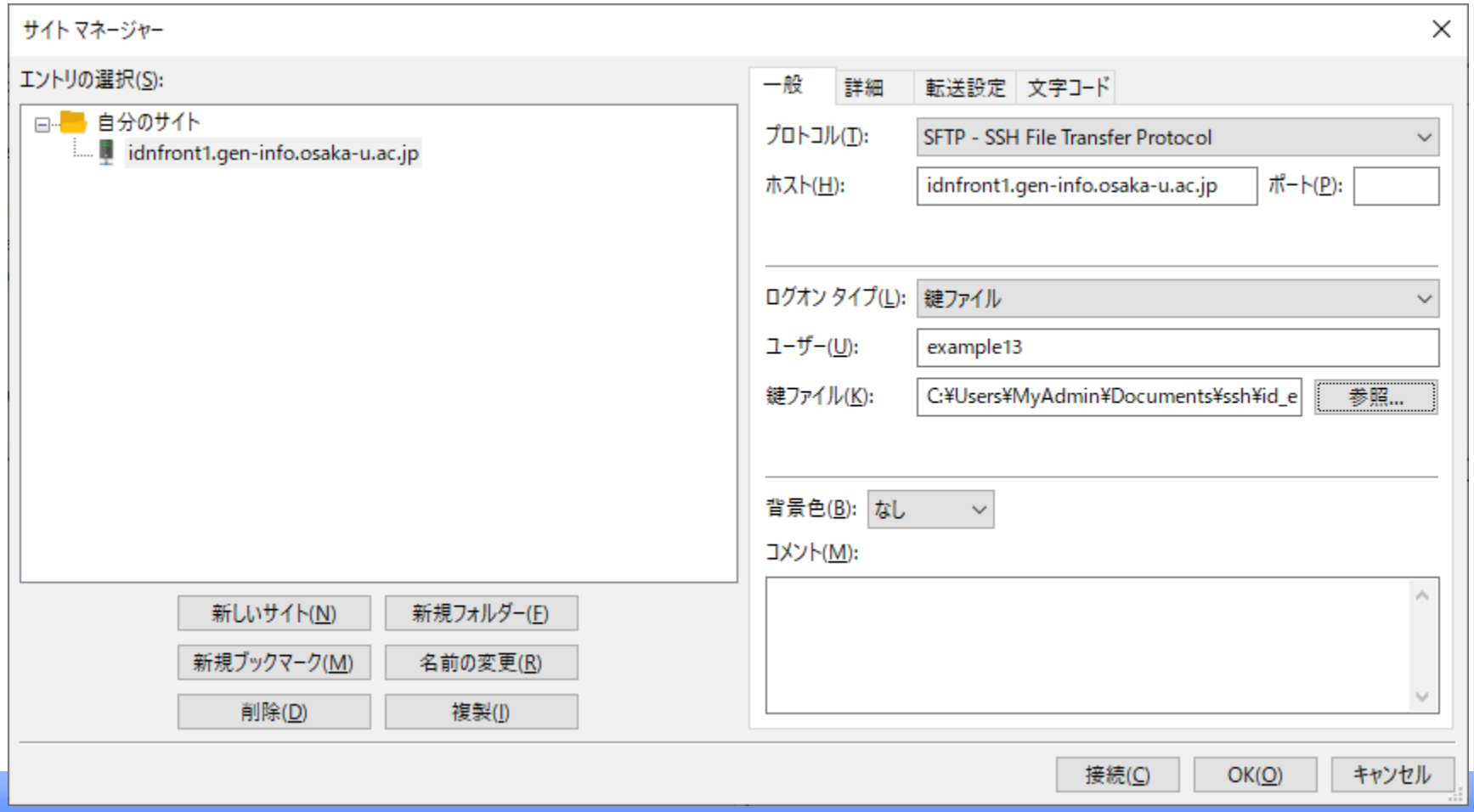
FileZillaの初期設定

- 「ファイル(F)」→「サイトマネージャー(S)」



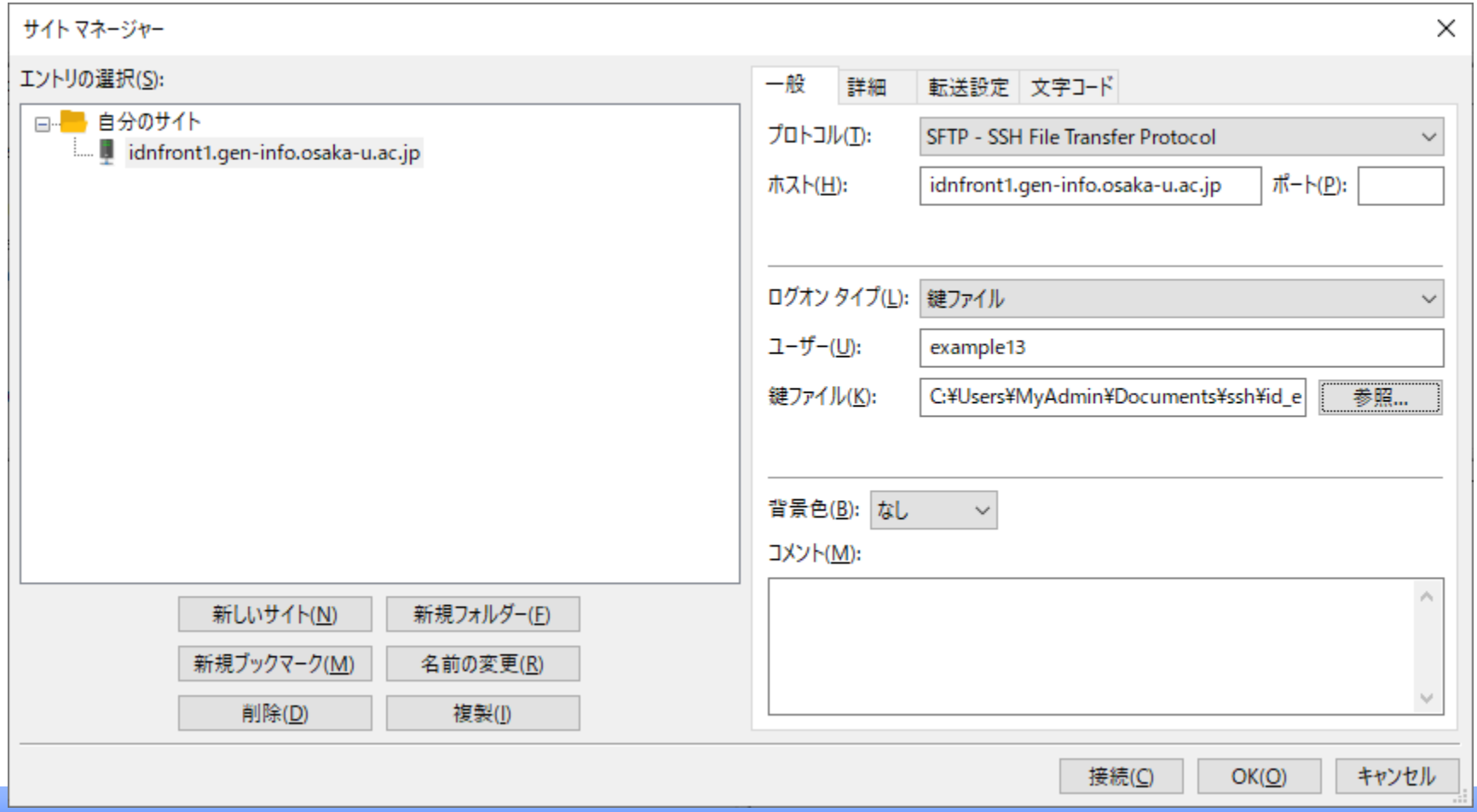
FileZillaの初期設定

- 「新しいサイト(N)」をクリック
- 「自分のサイト」の下の「新規サイト」:「idnfront1.gen-info.osaka-u.ac.jp」に変更
- 「プロトコル(T)」:「SFTPS – SSH File Transfer Protocol」に変更
- 「ホスト(H)」:「idnfront1.gen-info.osaka-u.ac.jp」を入力



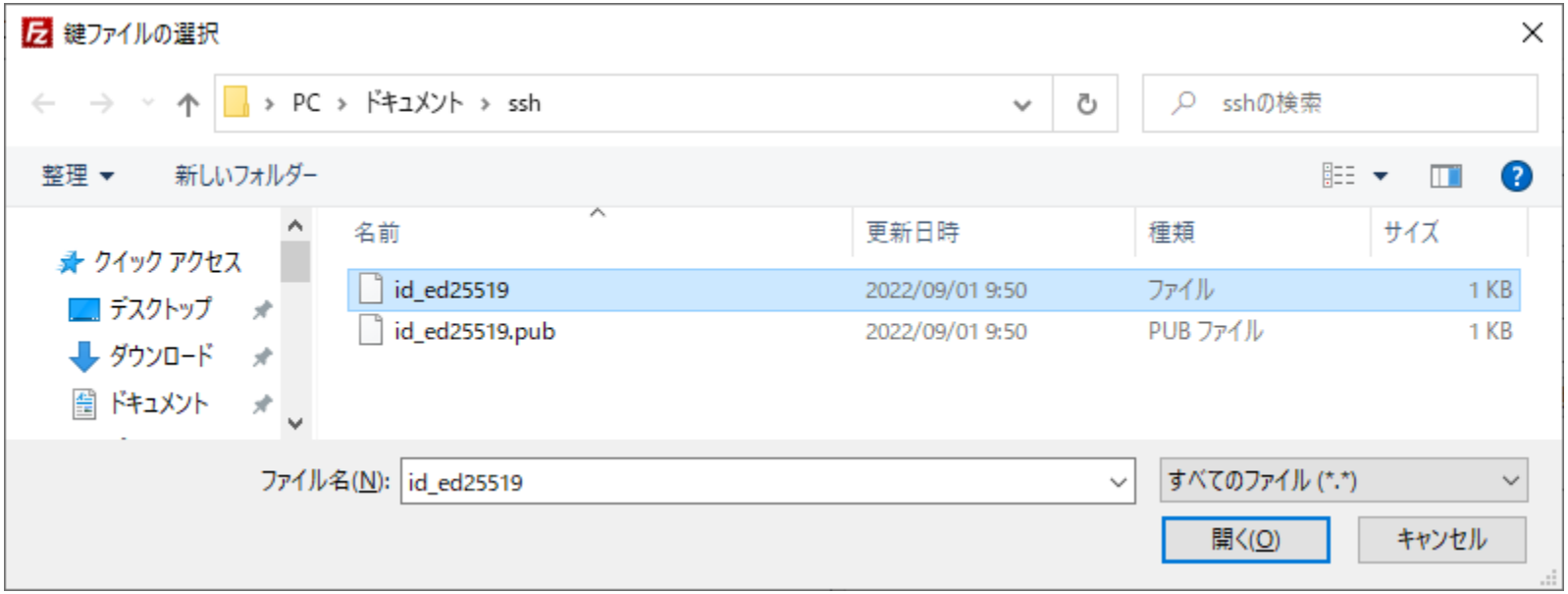
FileZillaの初期設定

- 「ログオンタイプ(L)」:鍵を作成済の場合は「鍵ファイル」に変更
  鍵を未作成なら「パスワードを尋ねる」に変更
- 「ユーザー(U)」:自分のIDを入力
- 「鍵ファイル(K)」:ログオンタイプが「鍵ファイル」の場合は「参照」をクリック。



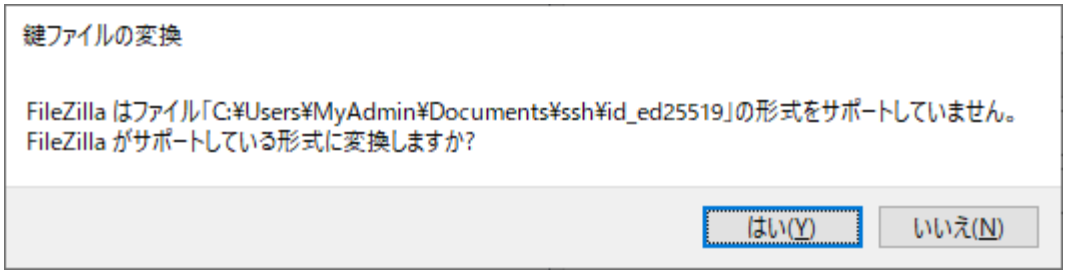
FileZillaの初期設定

- 「参照」をクリックすると「鍵ファイルの選択」画面が出る
- TeraTermで作成した鍵ファイル保存場所の「ドキュメント」→「ssh」を選択
- 「ファイル名(N)」の右側のプルダウンメニューを「すべてのファイル(*.*)」に変更
- 「id_ed25519」を選択し「開く(O)」をクリック

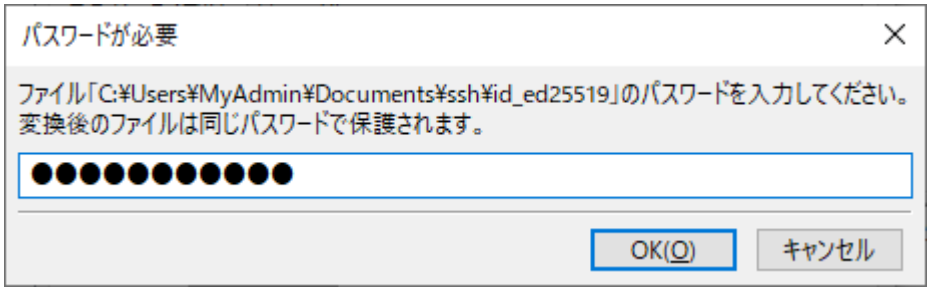


FileZillaの初期設定

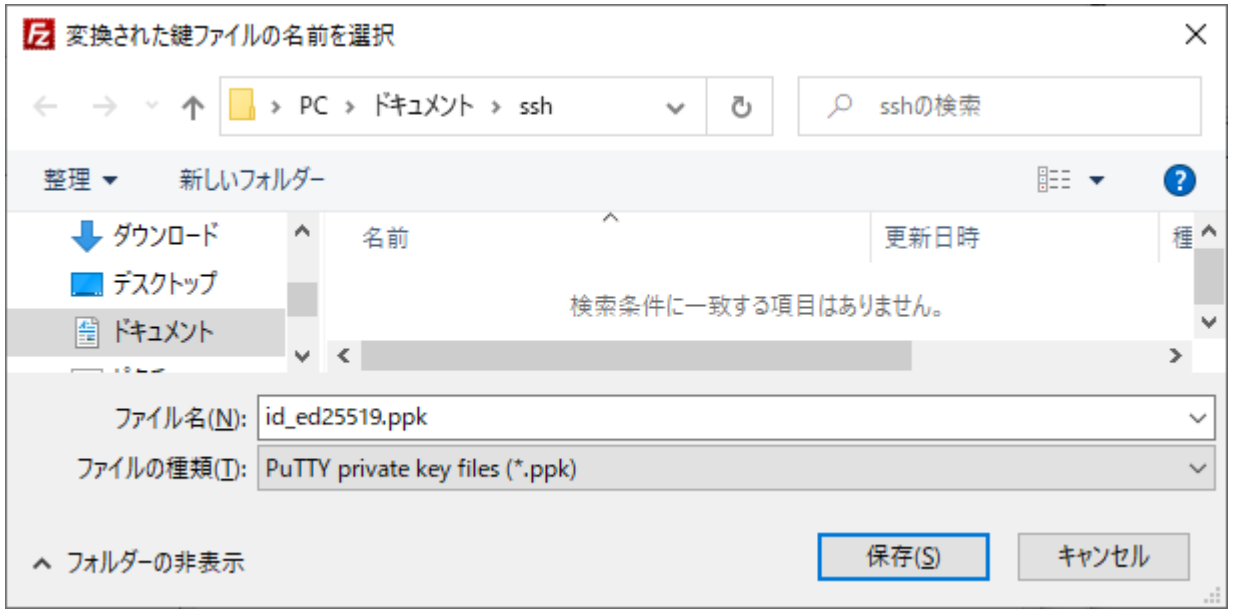
- 「鍵ファイルの変換」では
「はい(Y)」をクリック



- 「パスワードが必要」では
鍵ファイルのパスフレーズを
入力して「OK(O)」をクリック



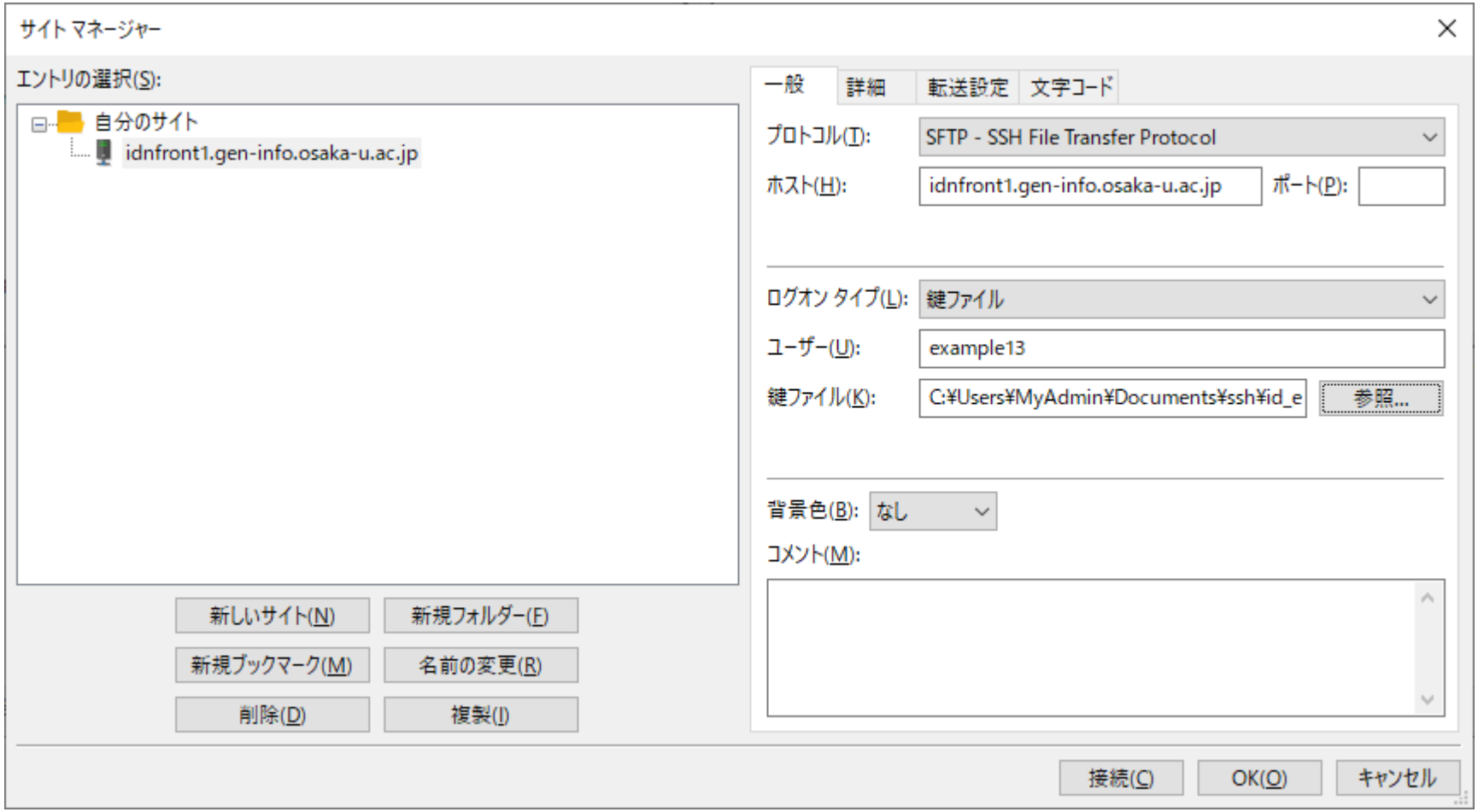
- 「変換された鍵ファイル
の名前を選択」では
「ファイル名(N)」に
「id_ed25519.ppk」
を手で入力して
「保存(S)」をクリック



- 「サイトマネージャー」の
画面に戻るので
「OK(O)」をクリックして
設定を保存

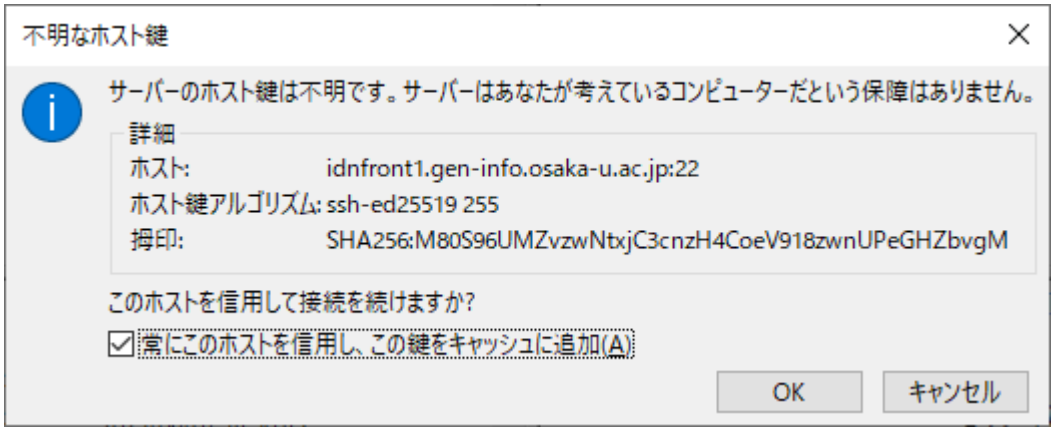
FileZillaによるSFTP接続

- 「ファイル(F)」→「サイトマネージャー(S)」を表示
- 「idnfront1.gen-info.osaka-u.ac.jp」を選択して「接続(C)」をクリック

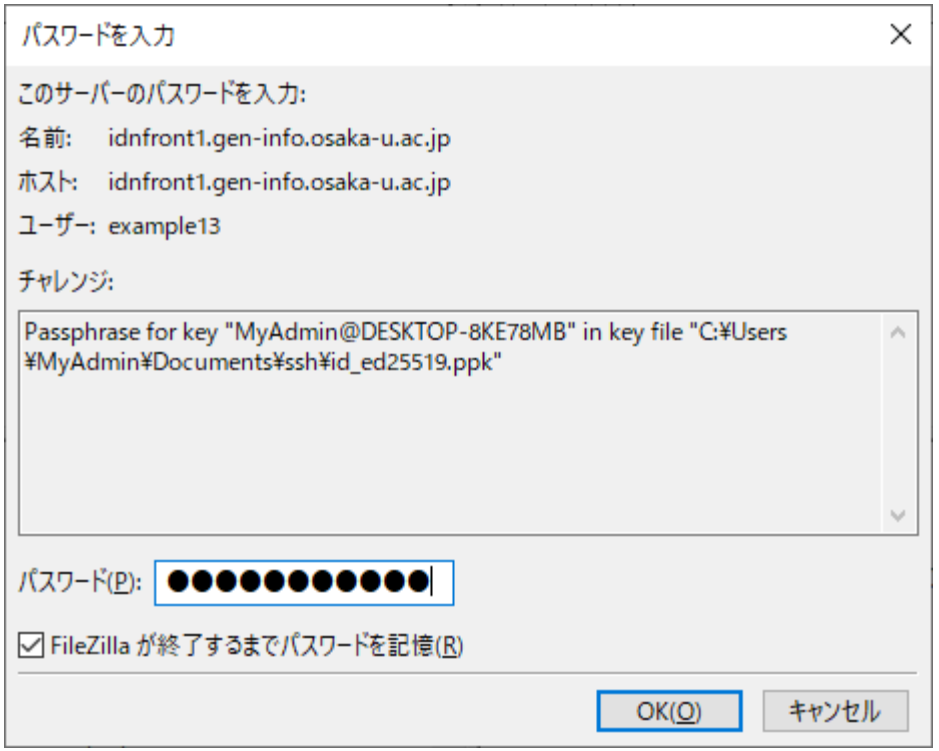


FileZillaによるSFTP接続

- 初回のみ「不明なホスト鍵」が出るので「常にこのホストを信用し、この鍵をキャッシュに追加(A)」にチェックを入れて「OK」をクリック

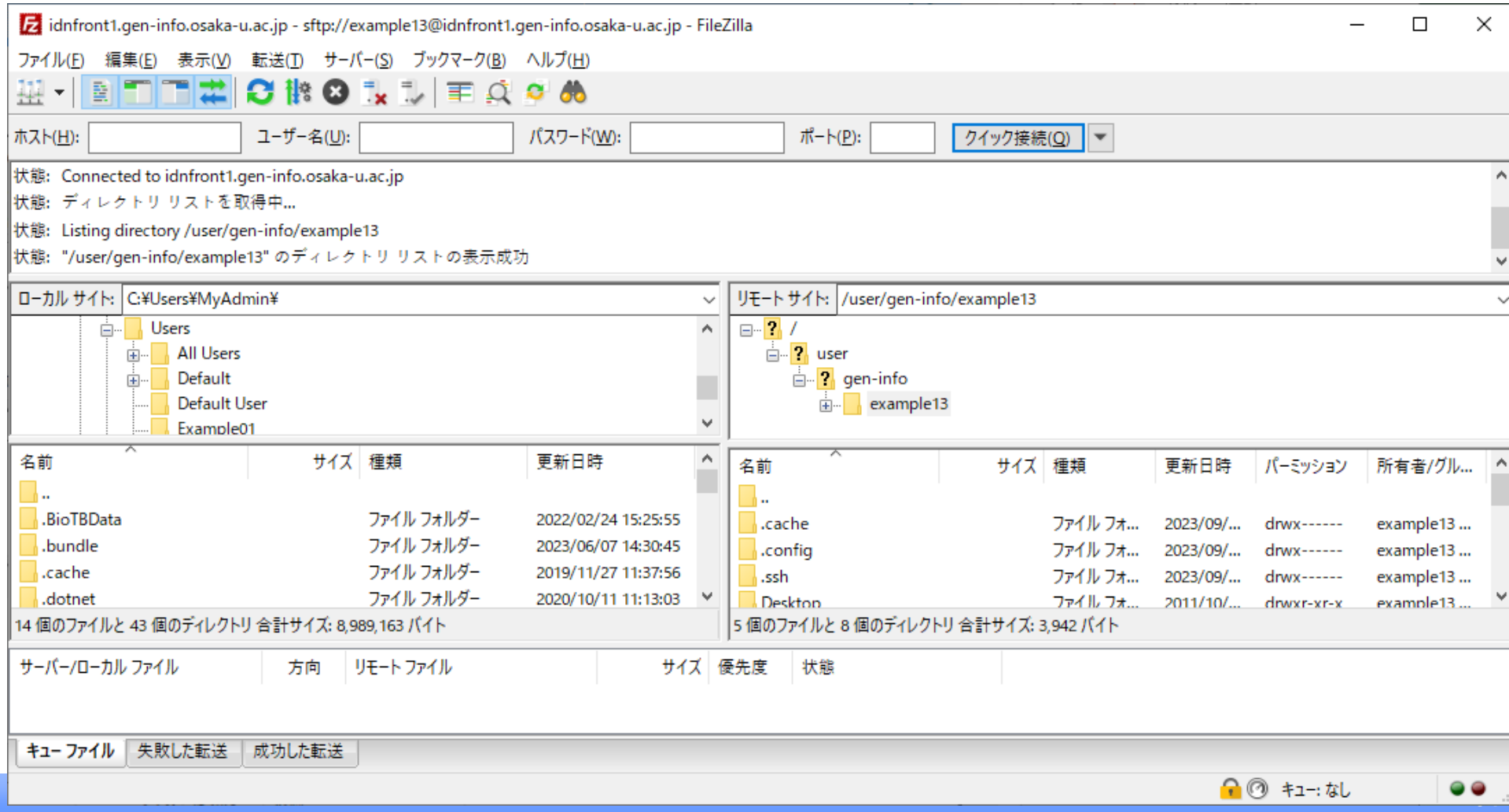


- 「パスワードを入力」が出る。
  ログオンタイプ「鍵ファイル」なら鍵ファイルのパスフレーズを入力。
  ログオンタイプ「パスワードを尋ねる」ならパスワードを入力。



FileZilla利用方法

- 左半分はパソコン側の、右半分はサーバー側のファイル一覧を表示
- 左→右や右→左でドラッグ＆ドロップしてファイル転送
- 右クリックして「アップロード」「ダウンロード」でもファイル転送できる



CLC Genomics Workbenchの利用

- CLC Genomics Workbench
  - QIAGEN社のNGS統合解析ソフト
  - マウス操作でNGSデータ解析が可能
  - マニュアルやチュートリアル等の資料が豊富
    - https://digitalinsights.qiagen.com/technical-support/manuals/
  - 有償ソフト：本センターは3ライセンス契約
    - CLC Genomics Server も利用可能
  - SSHポート転送+リモートデスクトップで使用可能

CLC Genomics Workbench利用方法

WindowsのTeraTermの場合

- TERATERM.INI内の[Hosts]部分を以下のように編集
  - Host1の行をコピーしてHost2（またはHost3など）の行を追加
  - その行の末尾に「 /ssh-L33389:127.0.0.1:3389」を追記

```
Host2=idnfront1.gen-info.osaka-u.ac.jp /ssh /auth=publickey /ask4passwd
/user=自分のID /keyfile=秘密鍵ファイルの絶対パス /ssh-L33389:127.0.0.1:3389
```

（注：表示の都合で折り返していますが、実際には1行で入力）
（注：アルファベット大文字小文字は厳密に区別します）

- 保存後、TeraTermを起動、上記ホスト設定でSSH接続
  - TeraTermのウインドウはCLC使用中は開いたままにする
  - 通常のコマンドライン操作に使っても構いません

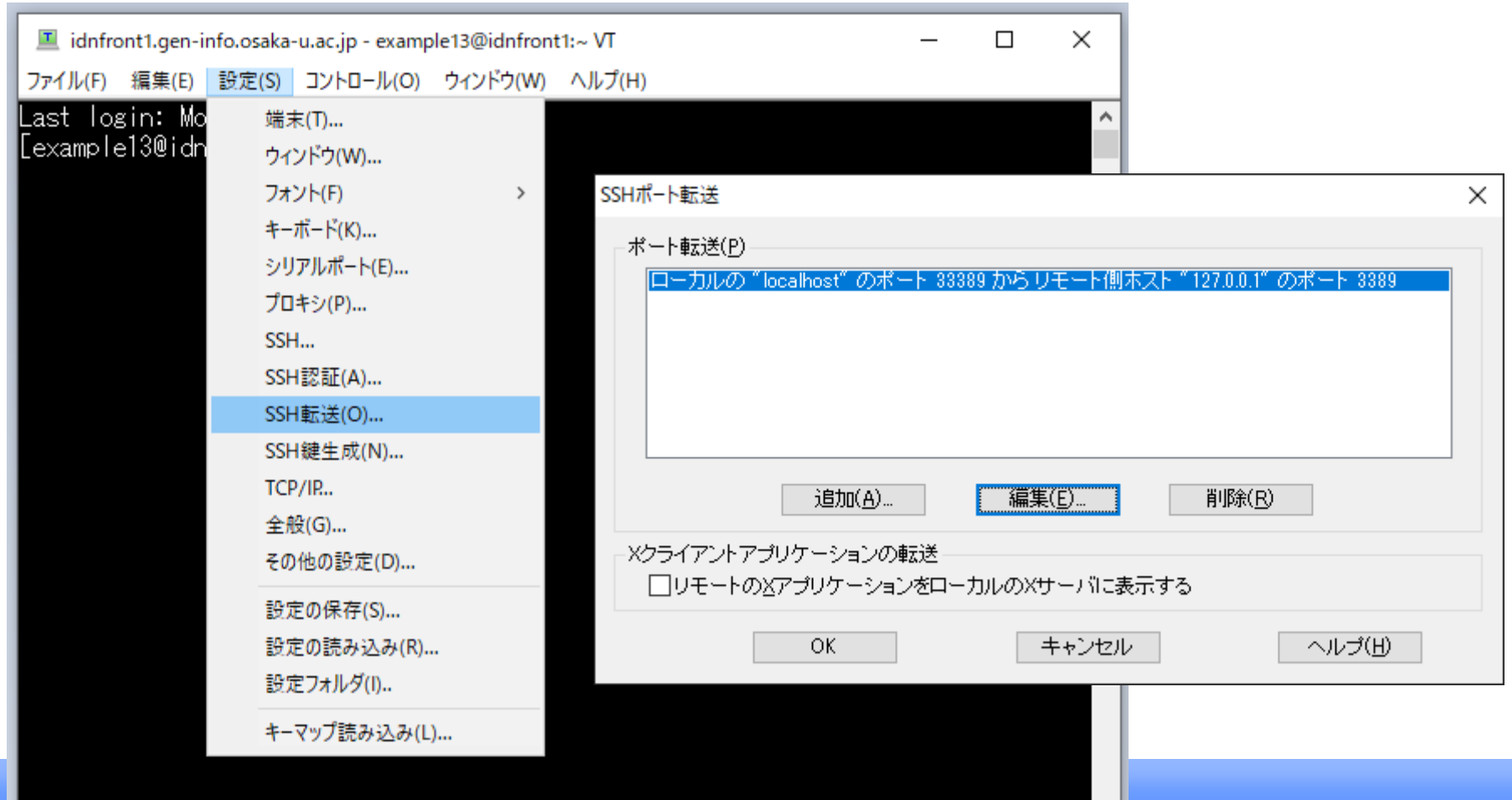
CLC Genomics Workbench利用方法

（参考：設定の確認方法）

「設定(S)」→「SSH転送(O)…」に以下の表示を確認

ローカルの “localhost” のポート 33389 からリモート側ホスト “127.0.0.1”のポート3389

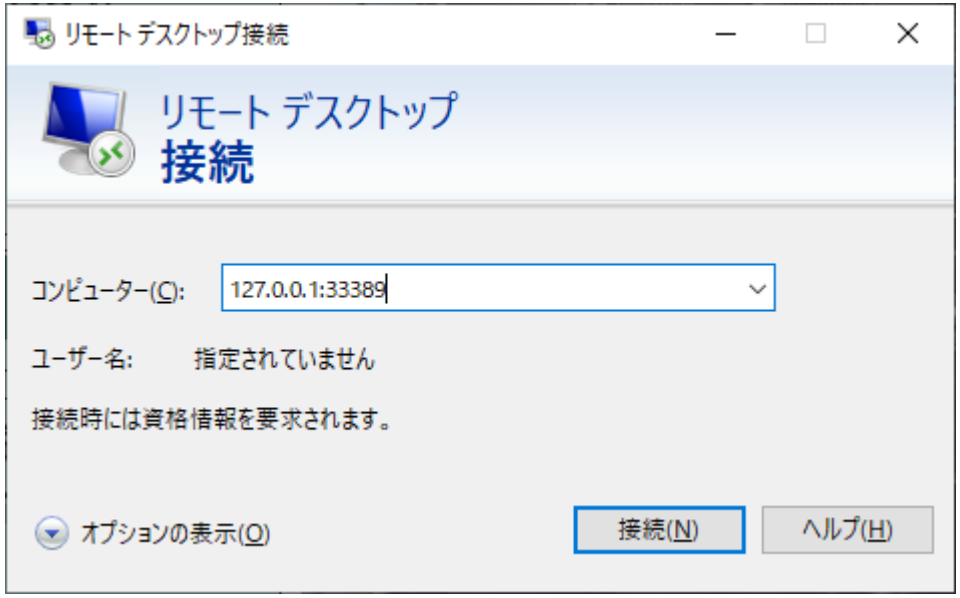
確認後「キャンセル」をクリック



CLC Genomics Workbench利用方法

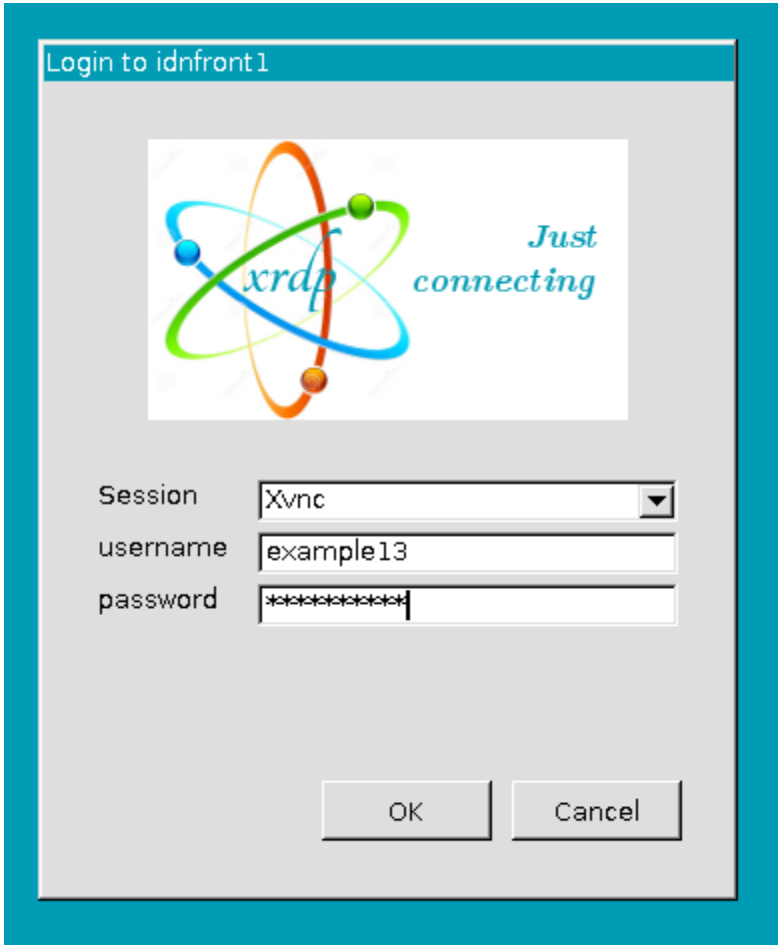
Windowsの「リモートデスクトップ接続」を起動

- タスクバーの検索ボックスに「mstsc.exe」と入力
  - または、スタートメニューの「W」→「Windowsアクセサリ」→「リモートデスクトップ接続」
- 「コンピュータ(C)」欄に「127.0.0.1:33389」を入力し「接続(N)」
- 右のような警告が出るが「はい(Y)」をクリックして続行



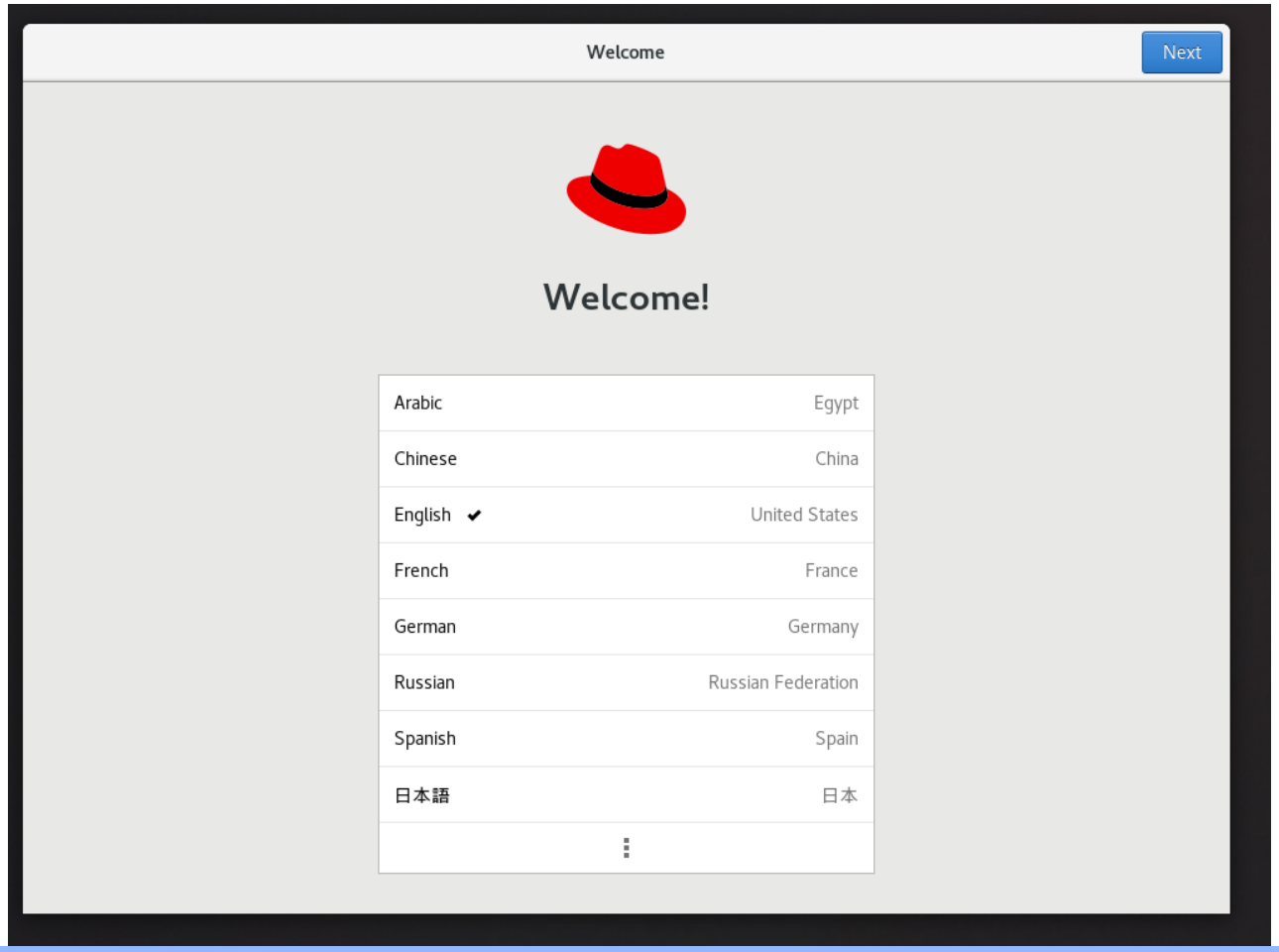
CLC Genomics Workbench利用方法

- 大きなリモートデスクトップ画面の中央に以下のログイン画面が出る
- 自分のIDとパスワードを入力して「OK」をクリック

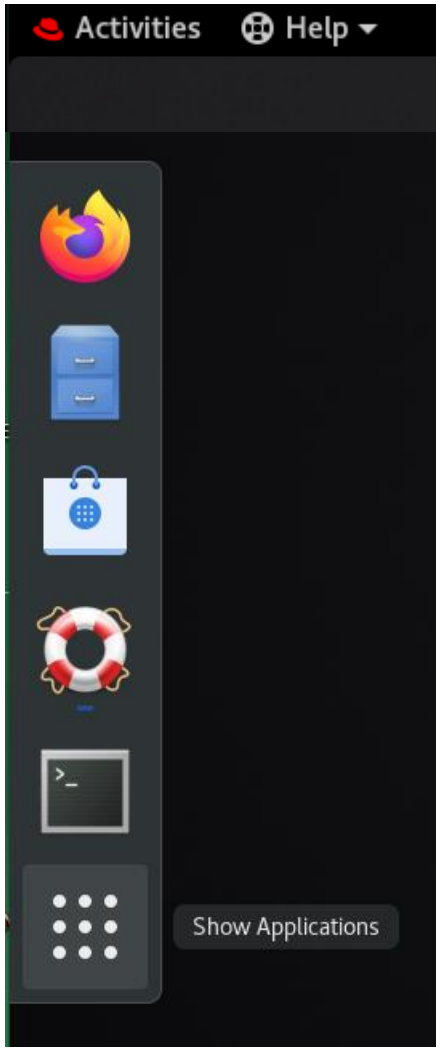


CLC Genomics Workbench利用方法

- 初回のみ言語などの設定画面が出る
- 「English」のままで構いません
- 途中の質問も「Next」または「Skip」で構いません



CLC Genomics Workbench利用方法

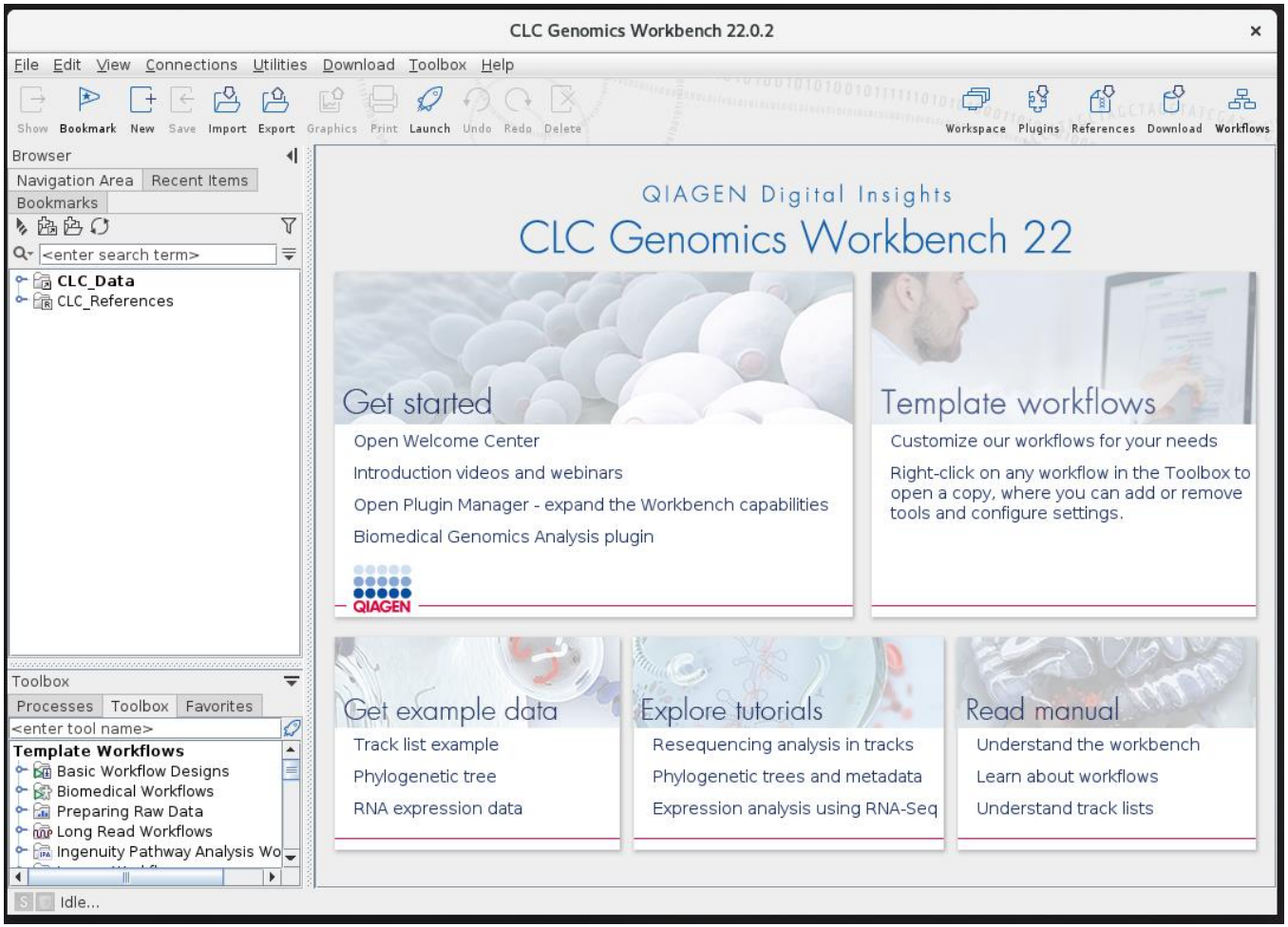


- 画面左上の「Activities」をクリック
- 点9個のアイコン（Show Applications）をクリック
- 画面中央上部の「CLC Genomics…」のアイコン
  クリックするとCLC Genomics Workbenchが起動

- （出ない場合は画面中央下部のボタンを「All」をクリック）

CLC Genomics Workbench利用方法

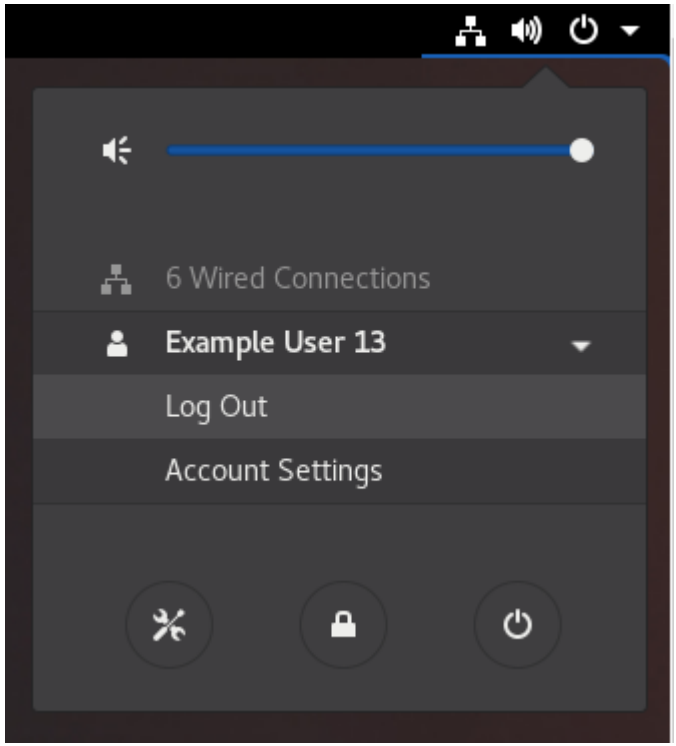
- 起動後の操作方法はマニュアル等の資料をご参照ください
- CLC Genomic ServerのGridを利用すると高速計算可能
  - 詳細はお問い合わせください



CLC Genomics Workbench終了方法

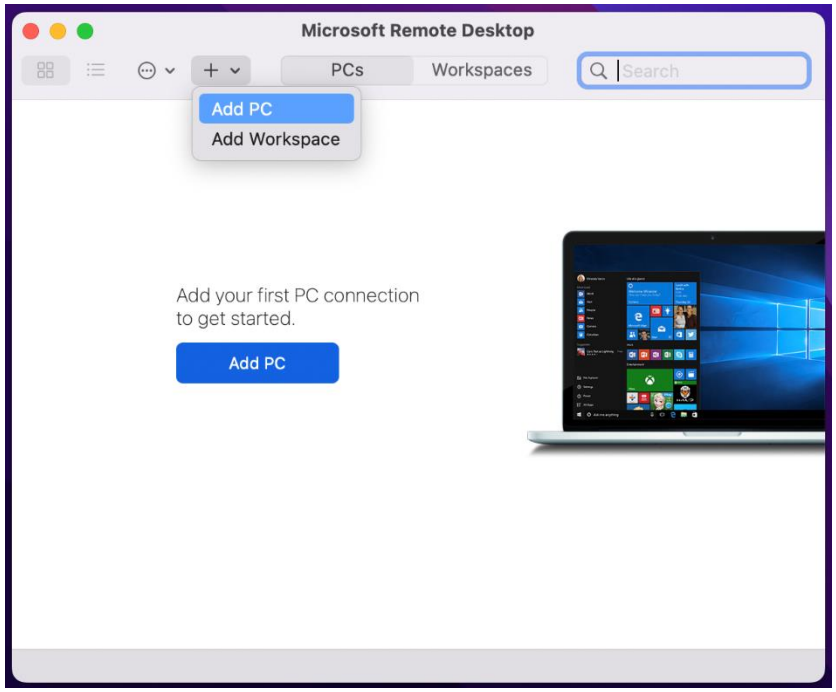
ライセンス数が限られるため、必ず終了操作を実施ください

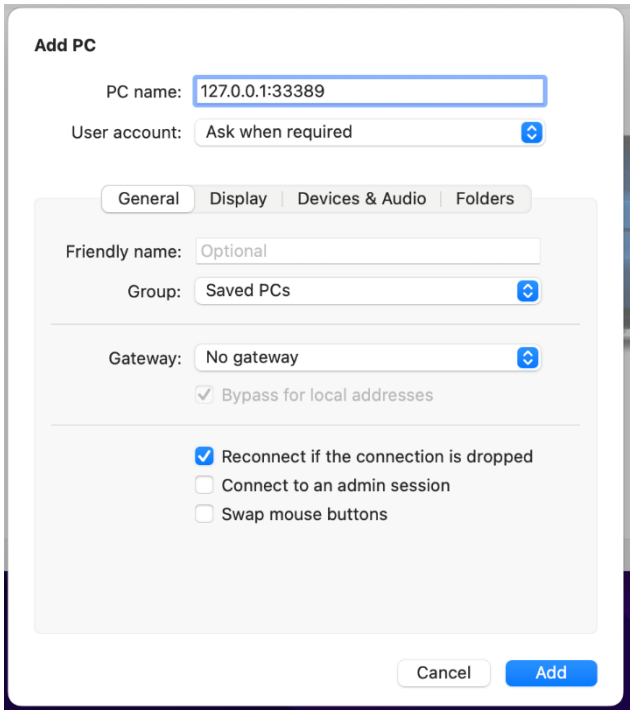
- CLC Genomics Workbench を終了
- 画面右上の電源ボタンアイコンをクリック
- 自分のIDの文字をクリック
- 「Log Out」をクリック
- 確認ウインドウが出るので「Log Out」を選ぶ
- 「リモートデスクトップ接続」接続前の画面に戻るので、「×」ボタンをクリックして終了
- TeraTermまたはターミナルにて「exit」と入力するなどしてSSH接続を終了



Macの場合：事前準備

- Microsoft Remote Desktopをインストール
  https://apps.apple.com/jp/app/microsoft-remote-desktop/id1295203466?mt=12
- Microsoft Remote Desktopを開く
- 「Add PC」をクリック
- 「PC Name」に「127.0.0.1:33389」を入力し「Add」をクリック



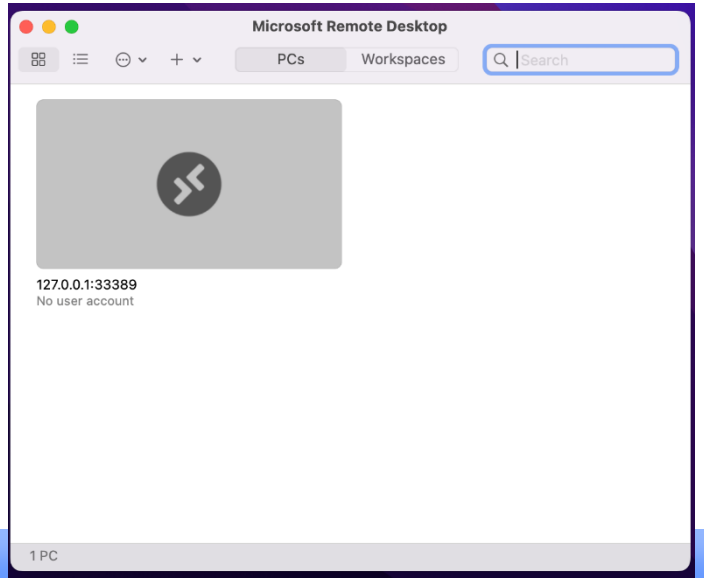


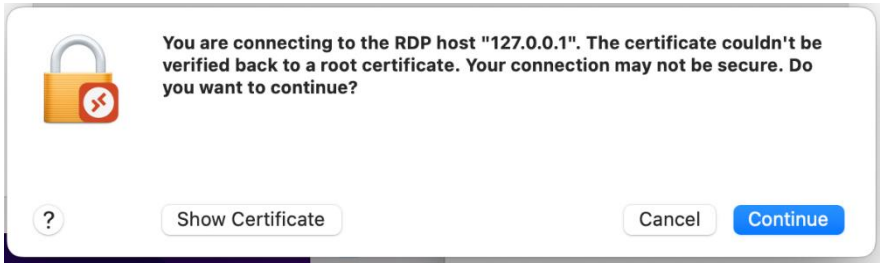
Macの場合：接続

- ターミナルの新規シェルのウインドウを作成し、以下のコマンドを入力
  （「自分のID」の部分はご自身のIDに置き換えてください）

```
ssh -L 33389:127.0.0.1:3389 自分のID@idnfront1.gen-info.osaka-u.ac.jp
```

- Microsoft Remote Desktopを起動
- 「127.0.0.1:33389」のアイコンをダブルクリック
- 「Username」に自分のID、「Password」にパスワードを入力し「Continue」
- 右図の警告が出るが「Continue」すると接続完了
- その後はWindowsの場合と全く同じ（終了操作も同じ）





計算機システム　サーバー機器

遺伝子解析サーバー

HPE SuperDome Flex
CPU: Intel Xeon Gold 6252 (2.1 GHz)×16
(合計384コア) (15.97 TFLOPS*)
(* AVX-512ベースコア周波数1.3GHz時の数値)
メモリ: 12 TB
OS: RedHat Enterprise Linux 8

フロントエンドサーバー

（ログイン・ジョブ管理用のサーバー）
HPE ProLiant DL360 Gen10 (2台)
CPU: Intel Xeon Gold 5220×2 (合計36コア)
メモリ: 256GB
OS: RedHat Enterprise Linux 8

ストレージ（ファイルサーバー）

DELL/EMC
PowerScale A2000 (16台) + F200 (4台)
物理容量: 5.1 PB (16TB × 320台)
実効容量: 4 PB (= 4,000 TB)

計算機システムの使い方

- ログイン先はフロントエンドサーバー
  - idnfront1.gen-info.osaka-u.ac.jp（通常利用）
  - idnfront2.gen-info.osaka-u.ac.jp（予備）
- 遺伝子解析サーバー（HPE Superdome flex）は強力だが直接ログインはできない
  - ジョブ管理システムPBS経由で利用

- 計算内容に応じて使い分ける
  - 軽い、短時間で終わるコマンド
    - フロントエンドサーバーで直接実行
  - 重い、時間のかかるコマンド
    - PBS経由にて遺伝子解析サーバーで実行

ジョブ管理システムPBS

- 正式名称：Altair PBS Professional
  - Altair社製のジョブ管理システム
  - NASA開発のPortable Batch System（PBS）を元に商用化

- 「多くのユーザが同じ計算サーバを共有し、メモリやCPUなどのリソースなどを公平に利用するためのシステム」
  （引用元：https://www.scl.kyoto-u.ac.jp/Attention/batch_sys.html ）
  - ユーザーが投入したジョブを預かり、実行可能なタイミングで実行
    - ジョブが実行可能になるまで預かり続ける
  - ジョブ実行のタイミングは様々な制約を考慮して決定
    - 計算機リソースの空き状況
    - キュー（ジョブを預かる仮想的な場所）のリソース制限
    - ユーザー単位のリソース制限（同時実行可能数など）
    - 受付順（他の制約が同じなら先着順）

ジョブ管理システムPBS

- 「ジョブスクリプト」を書く
  - 以下の指令を記述
    - 割り当てたいCPU数やメモリ量
    - 実行したいコマンド（複数可）
  - ジョブスクリプトは毎回使い捨てが基本（上級者除く）
    - テンプレートを用意しデータのファイル名のみ変更などは工夫可能
  - 途中でユーザーの手動操作が必要なコマンドは通常は書けない
    - 手動操作を狙ったタイミングで反映させる方法が通常は無い
- 「qsub」コマンドでジョブを投入
  - 投入後は即座にログアウト可能
  - 実行に時間がかかるコマンドも安心して任せられる
- 「qstat」コマンドで実行状況を確認

ジョブ管理システムPBS

- ジョブの投入先：キュー（ジョブキューやバッチキューと呼ぶこともある）
- 本センターでは3種類のキューを用意（管理用・特定ソフト用キューは別途存在）
  - INT: 小規模ジョブ用（後述の「qsub –I」からの使用を想定）
  - JOB: 通常ジョブ用
  - LONG: 大規模・長時間ジョブ用

- ジョブスクリプトを書かない簡易な利用も可能
  - 「qsub –I」を実行
  - 遺伝子解析サーバーを即座に利用可能

- 詳細はホームページ記載のマニュアルをご覧ください
  http://www.gen-info.osaka-u.ac.jp/localdocs/2022system/usage-abstract.html
  （学内専用ページ：学外から接続の方はお問い合わせください）

ソフトウェアのインストール・バージョンアップ

- ある程度のソフトウェアはシステムにインストール済
  - ただし、ソフトウェアのバージョンは原則として固定
    - バージョンアップで挙動が変化する可能性があるため
- 追加インストール等の希望は随時受付中
  - 以下の場合は希望に沿えない点はご了承ください
    - インストール済の他ソフトとの干渉（コンフリクト）
    - システム全体への副作用など、管理上の問題
  - このため、お断りせざるを得ない場合もあります

Condaの薦め

- Condaとは？
  - Anaconda社の開発するソフトウェアパッケージ管理システム
  - AnacondaまたはMinicondaというソフトウェア集に内包されている
    - Anaconda：科学技術計算用のPython・R環境を提供。
      個人利用は無償（中規模以上の企業は有償）。
    - Miniconda: Anacondaから必要最小限の部分を抜き出したもの。
      フリーソフトウェア（企業規模に関わらず無償利用可能）。

  - 一般ユーザー権限で動作。システム管理者権限不要。
    - 自分の必要なソフトの必要なバージョンを自由にインストール可能
  - 環境切替機能により、同ソフトの複数バージョン共存も可能
  - コミュニティによるソフトウェアパッケージの提供が盛ん
    - Conda-Forge : 一般および科学技術計算用パッケージ
    - Bioconda : 生命科学用パッケージ

Minicondaのインストール

- バイオインフォマティクス用途にはMinicondaがおすすめ
  （大学所属ならAnacondaでも必要ディスク容量が多い点以外の問題はありません）

- Miniconda公式サイトの「Linux」の記載を見てインストール
  https://docs.conda.io/projects/miniconda/en/latest/
  - インストール方法はときどき変更されるため、
    公式ドキュメントの最新の記載に従ってください

- 次回ログイン時からMinoconda環境が利用可能になる
  - 別ウインドウを開くか、いったんログアウトして再度ログイン

Biocondaの有効化

- Biocondaとは？
  - Conda用のソフトウェアリポジトリのひとつ
  - バイオインフォマティクス関連ソフトを多数収録
  - https://bioconda.github.io/
  - 公式サイトのUsageの記載を元に有効化
    - 方法は毎年のように変わるため要注意の旨の記載あり
  - たとえば「conda install samtools」など、コマンド1つでバイオインフォマティクス関連ソフトをインストールできる
    - パッケージの提供状況は https://anaconda.org/ の「Search Packages」から検索可能

- 参考ウェブサイト:
  大村 保貴（2023）『Bioconda 入門 Anaconda、Minicondaとの違いと EC2 に実行環境をセットアップしてみた』
  https://dev.classmethod.jp/articles/introduction-to-bioconda/

Condaの注意点

- Condaの欠点：環境が壊れやすい
  - 壊れる=パッケージのインストールやアップデートを行おうとした時にエラーが発生して進まなくなる
  - 壊れた仮想環境は放棄して別仮想環境を作成する方が簡単
    - エラーへの対処は高度な知識が必要のため
    - Miniconda丸ごと消して再インストールの方が楽な可能性すらある
  - Condaの仮想環境構築機能の活用が推奨
    - デフォルトの仮想環境（base）以外に複数の仮想環境を作成

DockerとSingularity

- Docker
  - 「コンテナ」(container)を作成・実行するソフトウェア
  - Docker社が開発
  - Linux版はフリーソフトウェア(Windows/Mac版は個人等無償)
- 「コンテナ」とは？
  - ソフトウェアとその周辺の実行環境(LinuxのOS部分を含む)を同時に封じ込めた環境
  - 配布時の周辺環境そのままで、そのソフトウェアを実行
    - インストールや環境整備の手間不要で、即座にソフトを使用開始できる
  - コンテナ内のOS(より正確にはカーネル)はLinux限定
    - RedHat, Ubuntu, Debianなど、ディストリビューションは問わない
  - 環境間の差分のみ保存する技術により、コンテナのファイルサイズを小さく保てる
  - Docker社が最初に開発、その後、標準規格化

Singularity

- Dockerの欠点：実行時に管理者(root)権限が必要
  - このため、本センターでは、Dockerは一般ユーザーには未提供
  - （比較的最近、管理者権限不要のrootless機能が追加されたが、前提条件が厳しいため、本センターでは使用不能）
- Singularity
  - HPC（High-Performance Computing）環境向けコンテナ実行環境
  - 管理者権限不要：一般ユーザー権限でコンテナを実行可能
  - Sylabs社が開発・主要部分はオープンソース（無償利用可能）
- Singularityは本センターにインストール済

  注意点：
  - Dockerでは動作するがSingularityでは動作しないコンテナがある
  - コンテナ作成には管理者(root)権限が必要のため、本センター環境ではコンテナ作成はできない（実行はできる）

例: Singularity上のRstudioでscRNA解析

解析部分の参考文献: https://labo-code.com/bioinformatics/seurat-1/

Rstudioコンテナイメージの取得

```
singularity pull docker://rocker/rstudio
```

コンテナ内に不足するUbuntu Jammy用の共有ライブラリの準備

```
（複雑のため省略）
```

以下は実際にはジョブスクリプト等に記載

qsub -Iコマンドで遺伝子解析サーバーにログイン

```
qsub -I -l select=1:ncpus=4:mem=384g
```

環境変数の設定

```
export SINGULARITYENV_USER=$USER
export SINGULARITYENV_PASSWORD=test0912
```

注：パスワードはこのRstudio Server接続専用とはいえ、毎回、変更が必要

例: Singularity上のRstudioでscRNA解析

テンポラリファイル置き場の準備

```
mytmp=/tmp/$(whoami)
mkdir -p $mytmp
chmod go-rwx $mytmp
mkdir -p $mytmp/tmp
mkdir -p $mytmp/run
mkdir -p $mytmp/var/lib/rstudio-server
```

Singularityの実行

```
singularity run ¥
--env LD_LIBRARY_PATH=$HOME/Ubuntu/jammy/usr/lib/x86_64-linux-gnu ¥
-B $mytmp/tmp:/tmp ¥
-B $mytmp/run:/run ¥
-B $mytmp/var/lib/rstudio-server:/var/lib/rstudio-server ¥
docker://rocker/rstudio ¥
rserver --server-user=$(whoami) ¥
--auth-none=0 ¥
--auth-pam-helper-path=pam-helper ¥
--www-port=48787
```

例: Singularity上のRstudioでscRNA解析

sshポート転送の準備：以下はパソコンで実行

TeraTerm: TERATERM.INIを編集して、末尾に以下を追記したHost行を追加し接続

```
/ssh-L48787:172.16.200.11:48787
```

Mac等:以下のコマンドをパソコンのターミナル上で実行

```
ssh -L 48787:172.16.200.11:48787 自分のID@idnfront1.gen-info.osaka-u.ac.jp
```

scRNAのデータの準備：上記でSSH接続した端末で実行

```
mkdir scRNAseq
cd scRNAseq
wget データのURL
tar xvf pbmc3k_filtered_gene_bc_matrices.tar.gz
```

パソコンのウェブブラウザで http://127.0.0.1:48787/ を開く

参考文献: https://labo-code.com/bioinformatics/seurat-1/ の解析を行う

例: Singularity上のRstudioでscRNA解析

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

RStudio Server

127.0.0.1:48787

File Edit Code View Plots Session Build Debug Profile Tools Help

example13

Go to file/function | Addins | Project: (None)

Console | Terminal | Background Jobs

R 4.3.1 · ~/scRNAseq/

```
Please refer to R-spatial evolution reports for details, especially
https://r-spatial.org/r/2023/05/15/evolution4.html.
It may be desirable to make the sf package available;
package maintainers should consider adding sf to Suggests:.
The sp package is now running under evolution status 2
     (status 2 uses the sf package in place of rgdal)
Attaching SeuratObject
> library(patchwork)
> setwd("~/scRNAseq")
> getcwd()
Error in getcwd() : could not find function "getcwd"
> getwd()
[1] "/user/gen-info/example13/scRNAseq"
> pbmc.data <- Read10X(data.dir = "./filtered_gene_bc_matrices/hg19/")
> pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells
= 3, min.features = 200)
Warning: Feature names cannot have underscores ('_'), replacing with dashes
('-')
> pbmc
An object of class Seurat
13714 features across 2700 samples within 1 assay
Active assay: RNA (13714 features, 0 variable features)
> # MT-で始まる全ての遺伝子の集合をミトコンドリア遺伝子の集合として使用する
> pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
>
> VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol
= 3)
Warning message:
In grSoftVersion() :
  unable to load shared object '/usr/local/lib/R/modules//R_X11.so':
  libXt.so.6: cannot open shared object file: No such file or directory
>
```

Environment | History | Connections | Tutorial

Import Dataset | 727 MiB | List

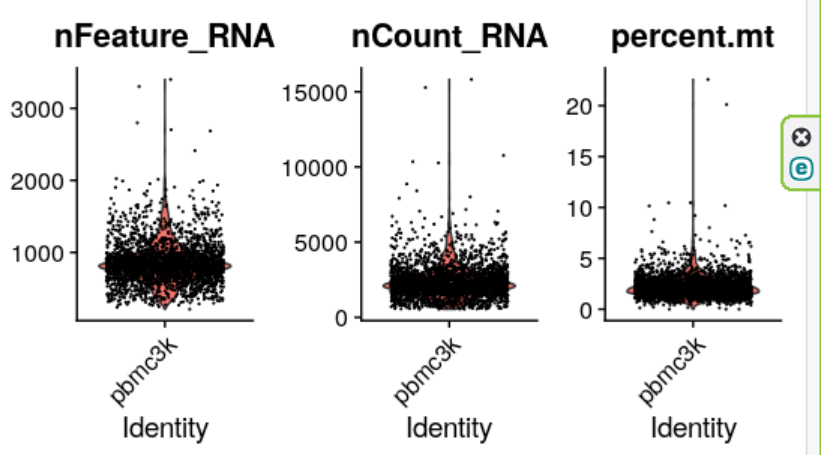
R | Global Environment

Data

pbmc	Large Seurat (58.5 MB)
pbmc.data	Large dgCMatrix (88392600 elements, 29.9 …

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



例: Singularity上のRstudioでscRNA解析

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

RStudio Server

127.0.0.1:48787

File Edit Code View Plots Session Build Debug Profile Tools Help

example13

Go to file/function | Addins | Project: (None)

Console | Terminal | Background Jobs

R 4.3.1 · ~/scRNAseq/

```
     (status 2 uses the sf package in place of rgdal)
Attaching SeuratObject
> library(patchwork)
> setwd("~/scRNAseq")
> getcwd()
Error in getcwd() : could not find function "getcwd"
> getwd()
[1] "/user/gen-info/example13/scRNAseq"
> pbmc.data <- Read10X(data.dir = "./filtered_gene_bc_matrices/hg19/")
> pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells
= 3, min.features = 200)
Warning: Feature names cannot have underscores ('_'), replacing with dashes
('-')
> pbmc
An object of class Seurat
13714 features across 2700 samples within 1 assay
Active assay: RNA (13714 features, 0 variable features)
> # MT-で始まる全ての遺伝子の集合をミトコンドリア遺伝子の集合として使用する
> pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
>
> VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol
= 3)
Warning message:
In grSoftVersion() :
  unable to load shared object '/usr/local/lib/R/modules//R_X11.so':
  libXt.so.6: cannot open shared object file: No such file or directory
> plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.m
t")
> plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_R
NA")
> plot1 + plot2
>
```

Environment | History | Connections | Tutorial

Import Dataset | 727 MiB | List

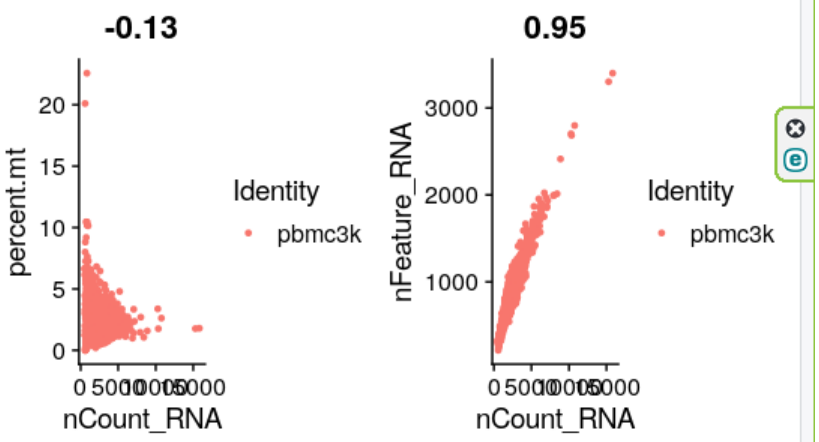
R | Global Environment

Data

pbmc	Large Seurat (58.5 MB)
pbmc.data	Large dgCMatrix (88392600 elements, 29.9 …
plot1	List of 9
plot2	List of 9

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish

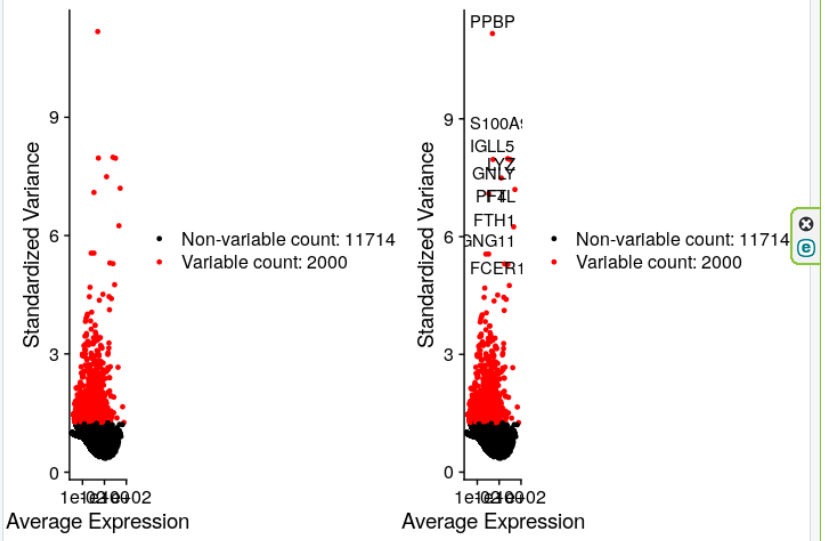


例: Singularity上のRstudioでscRNA解析

oubic

```
> 
> pbmc.data <- Read10X(data.dir = "./filtered_gene_bc_matrices/hg19/")
> pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)
Warning: Feature names cannot have underscores ('_'), replacing with dashes ('-')
> pbmc
An object of class Seurat
13714 features across 2700 samples within 1 assay
Active assay: RNA (13714 features, 0 variable features)
> # MT-で始まる全ての遺伝子の集合をミトコンドリア遺伝子の集合として使用する
> pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
> 
> 
> VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
> plot1 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "percent.mt")
> plot2 <- FeatureScatter(pbmc, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
> plot1 + plot2
> # LogNormalize
> pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize", scale.factor = 10000)
Performing log-normalization
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
> 
> # pbmcに再度入れる
> pbmc <- NormalizeData(pbmc)
Performing log-normalization
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
> # FindVariableFeatures関数を使ってデータを用意
> pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
Calculating gene variances
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
Calculating feature variances of standardized and clipped values
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
> 
> # top10の遺伝子名を記述のため、headを取得
> top10 <- head(VariableFeatures(pbmc), 10)
> 
> # プロット
> plot1 <- VariableFeaturePlot(pbmc)
> plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
When using repel, set xnudge and ynudge to 0 for optimal results
> plot1 + plot2
```

Environment	
Data	
pbmc	Large Seurat (59.2 MB)
pbmc.data	Large dgCMatrix (88392600 elements, 29.9 MB)
plot1	Large gg (9 elements, 1.3 MB)
plot2	Large gg (9 elements, 1.3 MB)
Values	
top10	chr [1:10] "PPBP" "S100A9" "IGLL5" "LYZ" "GNLY" "FTL" "PF4"…



例: Singularity上のRstudioでscRNA解析

- 課題
  - コンテナを使ってもなお複雑な手順が必要
  - docker://rocker/rstudioにはseuratに必要なライブラリが不足
    - 取り急ぎは、別途ダウンロードしたライブラリを LD_LIBRARY_PATH等で読み込む回避策で対応
    - 独自コンテナイメージを作成する、バグ報告またはpull requestを開発元に送って修正してもらう、などが根本的な解決策
  - ポート番号の48787は先着1名様のみ使用可能
    - 2人目以降は48787から別の数字に変更の必要あり
  - パスワード指定方法：このRstudio Serverへの接続専用ではあるが、毎回変更するのが好ましい

- 本センター計算機システムでRstudio, Jupyter Notebook等を簡単に実行可能な環境を提供する方法を検討中