Genomes Workflow - LBCM

01 - General guidelines and workflow organization

Wednesday 16th July, 2025

Contents

1 Overview

1.1 Module Objectives

This module covers:

Learning goals

- Basic Linux terminal concepts and usage.
- What is git and basic usage.
- Conda environment logic.
- Conda basic usage.
- Recommended bioinformatics project organization.

2 The Linux question

2.1

Key concepts:

Why this matters

- Important biological principle **example2024**
- Computational approach (**author2024**)
- Related methods **smith2024**

Definition 2.1. Key term or concept definition from **author2024**.

3 Tools & Software

3.1 Required Software

Installation notes

- **Primary tool:** Tool name and version
- **Dependencies:** Required libraries/packages
- **Optional:** Additional helpful tools

3.2 Installation Guide

```
# Installation commands
conda install -c bioconda tool_name
# or
sudo apt-get install package_name
```

Listing 1: Software installation

4 Workflow & Methods

4.1 Step-by-Step Protocol

Key parameters

1. **Data preparation:** Input requirements and formatting
2. **Quality control:** Initial data assessment

3. **Main analysis:** Core computational steps
4. **Result interpretation:** Output analysis and validation

Example 4.1. Practical example with real genomic data.

5 Practical Examples

5.1 Example 1: Basic Analysis

Input/output files

- **Input:** Sample data description
- **Command:** Based on approach from **example2024**
- **Output:** Expected results and file formats

```
# Example command with typical genomic data
tool_name -i input_file.fasta -o output_file.txt --parameter value
```

Listing 2: Basic command example

5.2 Example 2: Advanced Usage

Complex parameters

```
# Multi-step analysis pipeline
step1_tool input.fasta | step2_tool --param1 value1 > intermediate.txt
step3_tool intermediate.txt --param2 value2 -o final_result.txt
```

Listing 3: Advanced analysis pipeline

6 Results & Interpretation

6.1 Output Files

Common output formats and their interpretation:

File formats

- **Format 1:** Description and typical contents
- **Format 2:** When and how to use this output
- **Quality metrics:** How to assess result quality

Remark 6.1. Important note about result interpretation following **author2024**.

7 Scripts & Code

7.1 Helper Scripts

```
#!/usr/bin/env python3
"""
Helper script for genomic data processing
Usage: python script.py input.fasta output.txt
"""

def process_sequences(input_file, output_file):
    """Process genomic sequences"""
    with open(input_file, 'r') as f:
        sequences = f.read()

    # Processing logic here
    processed = sequences.upper()

    with open(output_file, 'w') as f:
        f.write(processed)

if __name__ == "__main__":
    import sys
    process_sequences(sys.argv[1], sys.argv[2])
```

Listing 4: Data processing script

7.2 Quality Control

```
#!/bin/bash
# Quality control pipeline for genomic data

# Check file format
file_format_check.py $INPUT_FILE

# Basic statistics
sequence_stats.py $INPUT_FILE > stats.txt

# Quality assessment
quality_assessment_tool $INPUT_FILE --output qc_report.html
```

Listing 5: QC pipeline

8 Troubleshooting & Best Practices

8.1 Common Issues

Error solutions

- **Memory errors:** Reduce dataset size or increase available RAM
- **Format issues:** Check input file formatting and encoding
- **Parameter tuning:** Guidelines for optimization

8.2 Best Practices

- **Data backup:** Always keep original data copies
- **Version control:** Track analysis versions and parameters
- **Documentation:** Record all analysis steps and decisions
- **Reproducibility:** Use consistent environments and seeds

9 Exercises & Next Steps

- **[TODO]: Practice with provided sample data**
- **[TODO]: Try different parameter settings**
- **Apply to your own genomic dataset**
- **[TODO]: Explore advanced features**

10 Research Notes

Additional observations and module-specific notes...

Key insight: Connection between this tool and genome annotation pipeline

[IDEA]: Extension: Integration with other bioinformatics tools in the workflow