Kinexus Bioinformatics - CytoNET BCIT 2025

This document is intended to explain the use and maintenance of CytoNET. The project was built as a separate website from the main Kinector site.

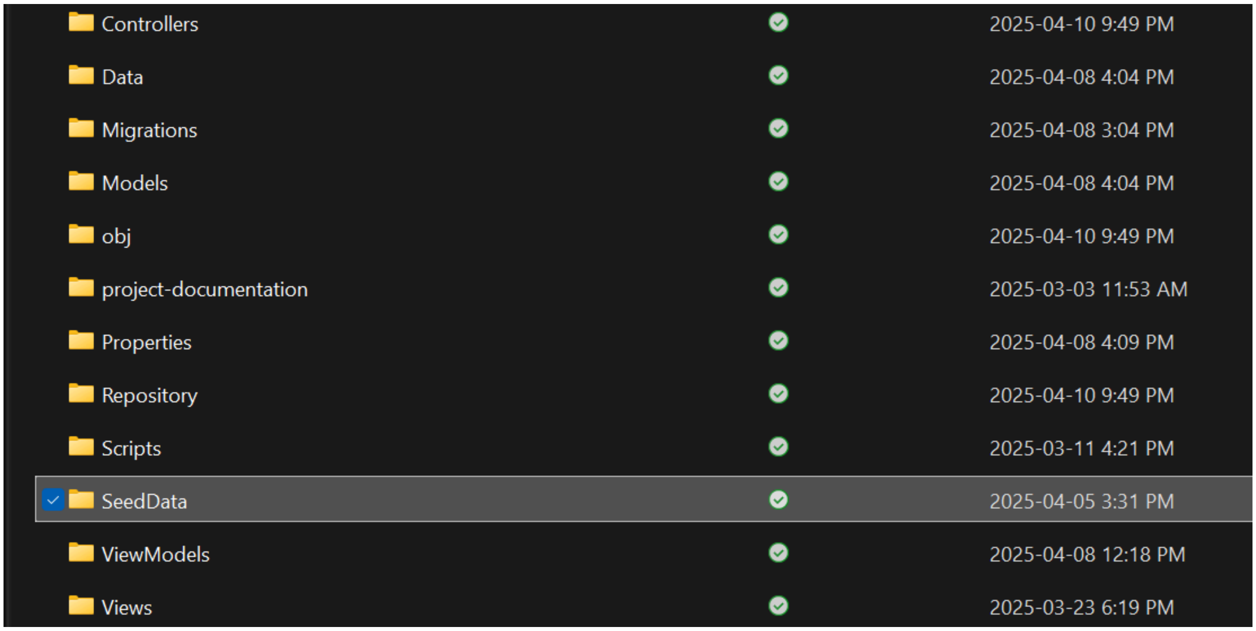
Assuming the site has already been hosted, the only things you'll need to do for site upkeep are adding new data. As it is set up now, the new data must be in the same format as the old ones. In other words, the column layout and names must be in the same order and format between old and new spreadsheets/csv's/Excel files.

Adding Code to Server

1. Unzip code folder into the desired location on the server
2. Open terminal inside of the main codebase folder
3. Run Commands to initialize the SQLite database:
    a. dotnet build
    b. ./Scripts/run-migrations.sh
4. Then use the command dotnet run to start the application, the databases will seed initially so it will take a while to start up.

Adding New Data

Inside the main project, the data from the excel spreadsheets is being stored in CSV files inside the SeedData folder.



Controllers	✓	2025-04-10 9:49 PM
Data	✓	2025-04-08 4:04 PM
Migrations	✓	2025-04-08 3:04 PM
Models	✓	2025-04-08 4:04 PM
obj	✓	2025-04-10 9:49 PM
project-documentation	✓	2025-03-03 11:53 AM
Properties	✓	2025-04-08 4:09 PM
Repository	✓	2025-04-10 9:49 PM
Scripts	✓	2025-03-11 4:21 PM
☑ SeedData	✓	2025-04-05 3:31 PM
ViewModels	✓	2025-04-08 12:18 PM
Views	✓	2025-03-23 6:19 PM

In this folder you will find the 5 files used for seeding the database. Each sheet on the shared excel document will have to be downloaded locally and converted to CSV format.

NOTE

CSV files and database storage do not allow for any text formatting. In TableD there is bolded text that must be appended with an asterisk (*) in order to be displayed in bold text on the website. We have created a macro that can do this for all the required cells.

The sheet must be saved from a shared XLSX file to a google sheet, then the following macro can be imported using the Extensions > Apps Script in the navigation menu. The Macro has been pasted at the end of this document, but will also be saved in the project folder under the Scripts folder.

Pase the macro into the script section and save it under some name, then navigate back to TableD and use Extensions > Macros > <your-macro-name> to run it. Once it has been completed you will get a notification, double check that the bolded cells have been appended with an asterisk (*).

You can now download the tables one by one and convert them to CSV format. Drag them into the project SeedData folder replacing the existing files. Once all files have been replaced, simply restart the server, or re-publish the application if using Azure to have the new data seeded into the database to be used on the live site.

```
function markBoldCellsWithAsterisk() {
  // Get the active spreadsheet and sheet
  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
  var sheet = spreadsheet.getActiveSheet();

  // Define column indices (1-based in Google Sheets)
  var startCol = 8;    // Column H is index 8
  var endCol = 28;     // Column AB is index 28
  var colCount = endCol - startCol + 1;  // 21 columns total

  // Find the last row with data in the sheet
  var lastRow = sheet.getLastRow();

  // If the last row is less than 3, there's no data to process below row 2
  if (lastRow < 3) {
    SpreadsheetApp.getUi().alert("No data found below row 2 in the specified columns.");
    return;
  }

  // Set the range to process (H3:AB[lastRow])
  var processRange = sheet.getRange(3, startCol, lastRow - 2, colCount);

  // Get the rich text objects to check for bold formatting
  var richTextValues = processRange.getRichTextValues();
  var values = processRange.getValues();
  var hasChanges = false;

  // Loop through each cell in the range
  for (var i = 0; i < richTextValues.length; i++) {
    for (var j = 0; j < richTextValues[i].length; j++) {
      // Skip empty cells
      if (values[i][j] !== "") {
        var richText = richTextValues[i][j];
        var text = richText.getText();

        // Skip if already ends with asterisk
        if (text.slice(-1) === "*") {
          continue;
        }

        // Check for bold formatting
        var isBold = false;
        var runs = richText.getRuns();

        // Check if any part of the text is bold
        for (var k = 0; k < runs.length; k++) {
          if (runs[k].getTextStyle().isBold()) {
            isBold = true;
            break;
          }
        }

        // If any part is bold, add asterisk
        if (isBold) {
          values[i][j] = text + "*";
          hasChanges = true;
        }
      }
    }
  }

  // Only update if changes were made
  if (hasChanges) {
    processRange.setValues(values);
  }

  // Show completion message
  SpreadsheetApp.getUi().alert("Process complete. Bold cells in columns H to AB (starting from row 3) have been marked with an asterisk (*)");
}
```

Technical Things

- As was previously mentioned, Windows Server 2008 is deprecated, meaning that it is no longer being maintained and will not be serviced by Microsoft any more. Upgrading to a newer version would mean better security (as the 2008 version doesn't receive security updates), easier for new teams to work with it and would prevent accumulation of further technical debt.
- Versions
  - The site is being hosted with Microsoft Azure using ASP.Net Core version 8.0
  - Using SQLite Version 9.0.1
    - SQLite is a lightweight version of mysql that runs locally, meaning it does not need to be hosted separately on your server and will be self-contained with the application itself.