User Manual

GC-AI

Introduction

- **Name:** GC-AI
- **Description:** GC-AI is a web-based tool designed to convert hand-drawn genetic circuit diagrams into the SBOL3.1 standard. Through image processing, it identifies biological components, defines their roles, and generates standardized documentation via an intuitive interface.

Access and Navigation

Accessing the Website

- **URL:** https://www……. (To be updated once the official domain is live).
- **Browser Requirements:** To ensure full functionality and optimal performance, GC-AI requires the following browser capabilities:
  - Supported Browsers:
    - Google Chrome (v61 or newer)
    - Mozilla Firefox (v60 or newer)
    - Microsoft Edge (v16 or newer)
    - Safari (v10.1 or newer)
    - Brave (latest version)
- Required Features:
  - JavaScript enabled
  - Support for modern web standards: ES6+, Fetch API, CSS Grid, and Flexbox

Local Testing Requirements

To run GC-AI locally, you need to have the following installed on your machine:

- Docker
- Docker Compose (already included in Docker Desktop)
- Git

For detailed installation instructions, refer to the local testing repository: https://github.com/cvtalrc/GC-AI-environment/

Navigation

- Two main views are provided:

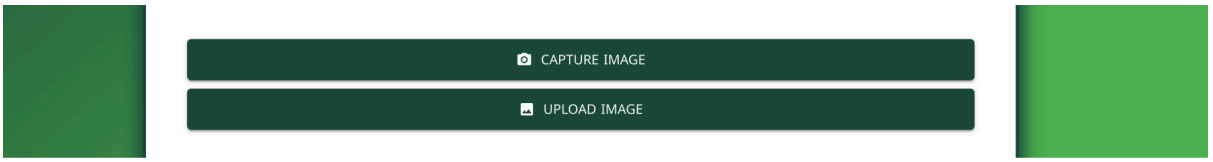
1. **Home and Image Upload View:** Allows users to upload or capture images.
2. **Configuration View:** Enables adjustments to the structural and functional settings of the system.

Step-by-Step Operation

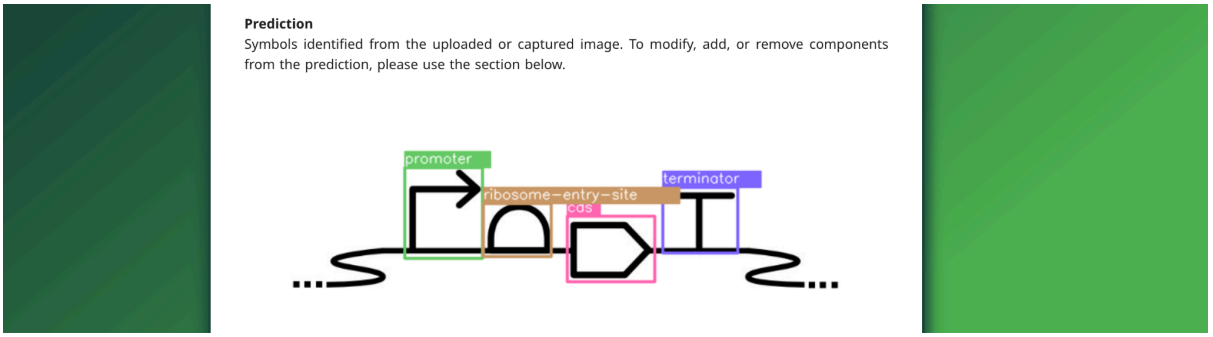
Step 1: Uploading or Capturing an Image

- **How to Proceed:** Upload an image from your device or capture it directly using the web page.

  Recommendation: For better inference, the image should only contain circuit components, without interactions or external elements such as proteins or simple chemicals. These can be added later in the process.

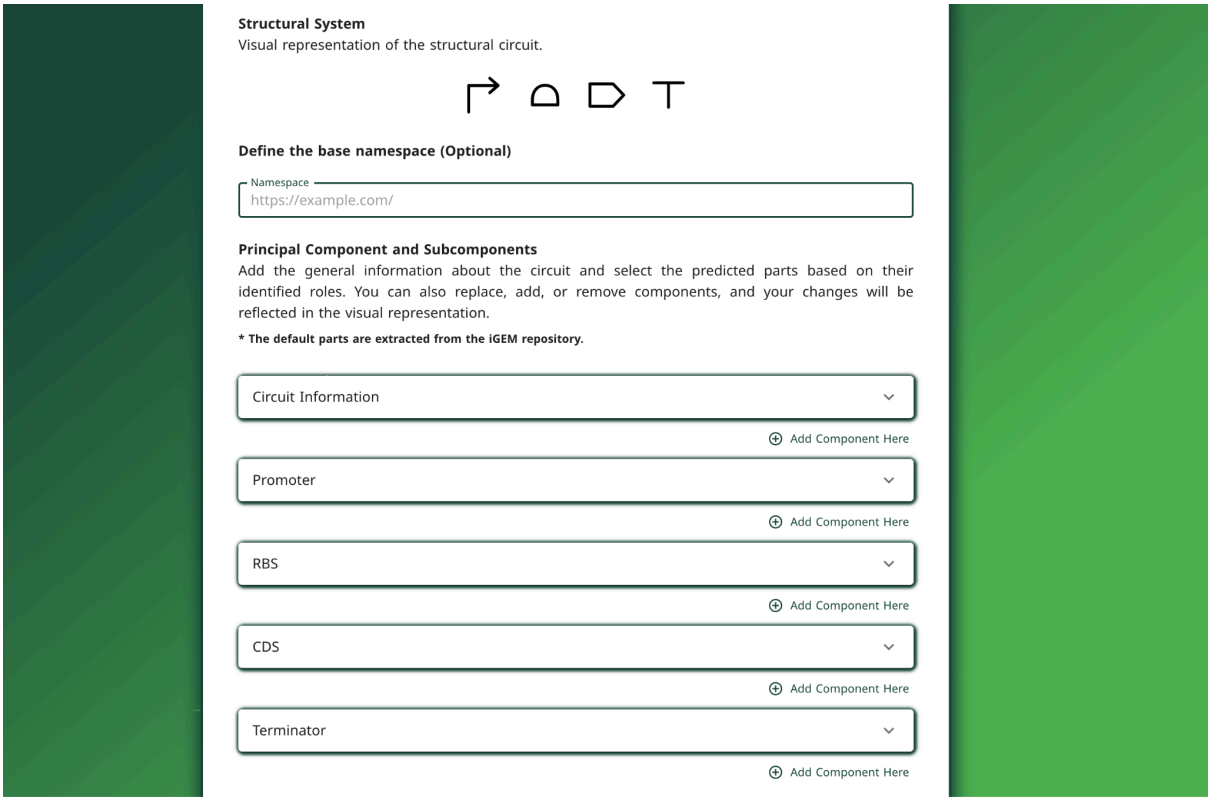


- **Processing:** The webpage processes the image automatically using an AI model (Faster R-CNN with ResNet50-FPN) to identify each part of the circuit.



Component Prediction

- **Automatic Prediction:** Each identified symbol generates an accordion menu with the component details. The predicted role is assigned by the model.
- **Manual Corrections:** Users can add, delete, or modify components directly in case of prediction errors.
- **Dynamic SVG Visualization:** The structural preview updates dynamically with each modification. The original predicted image remains constant.



Step 2: Workspace Configuration (Optional)

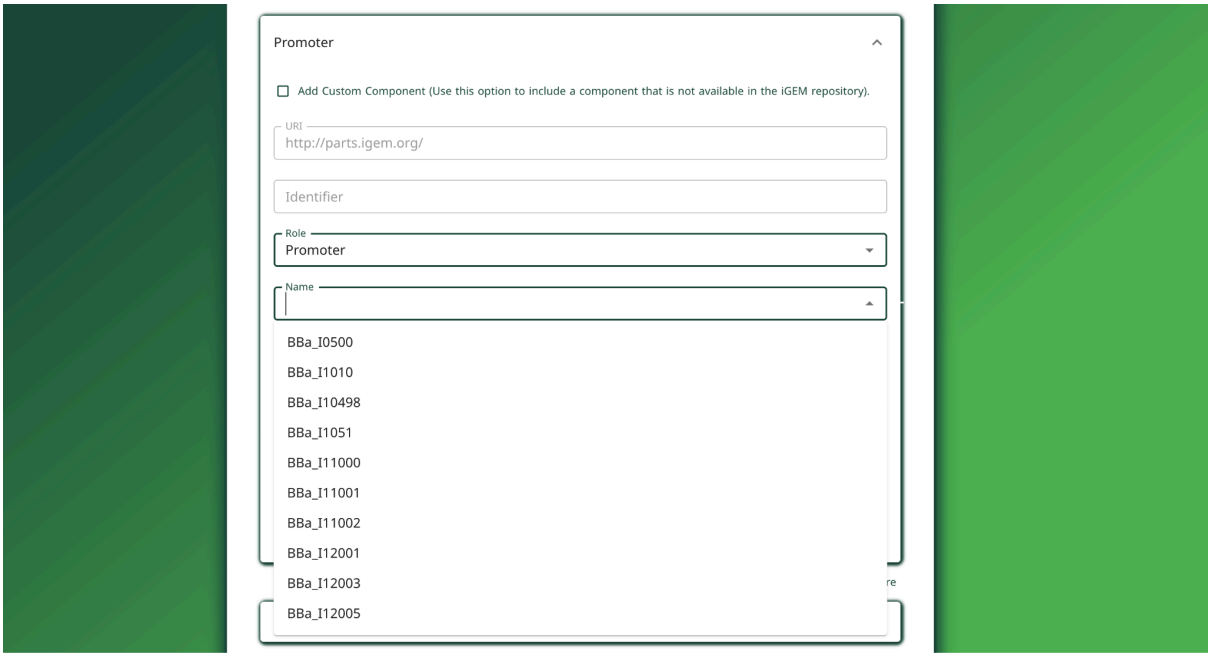
- **Namespace:** The namespace uniquely identifies all elements in the SBOL3.1 document.
  - By default, the current iGEM database (2024) is used.
  - Users can also specify a custom namespace.

Step 3: Structural System Configuration

- **Selecting Components via iGEM:** Specific components can be selected using the dropdown in the 'name' field.

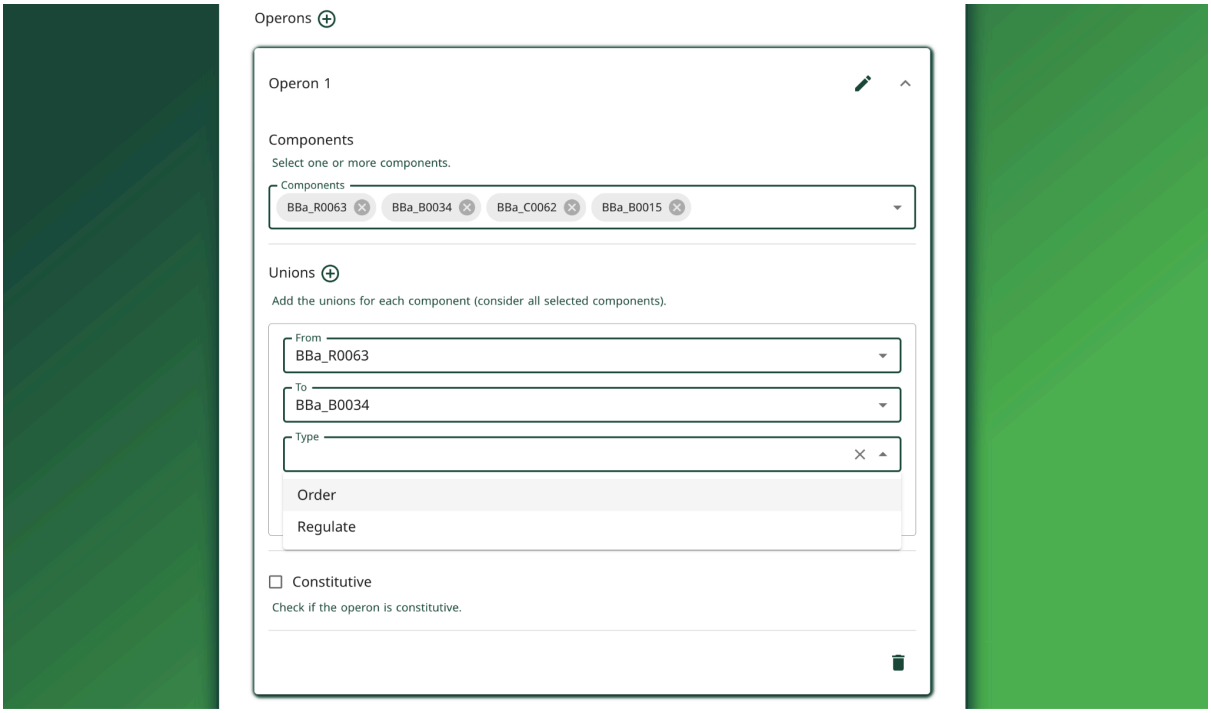
- **Creating Custom Components:** If a component is not available in iGEM, users can create it by specifying fields such as URI, role, name, and sequence.

- **Sequence Generation:** All connections between components are of the "order" type to correctly generate the circuit.

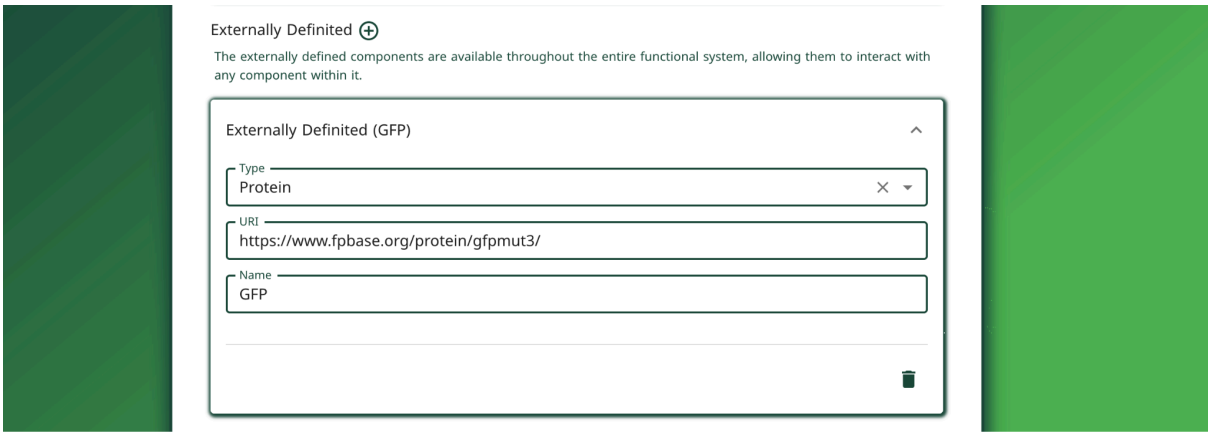


Step 4: Functional System Configuration (Optional)

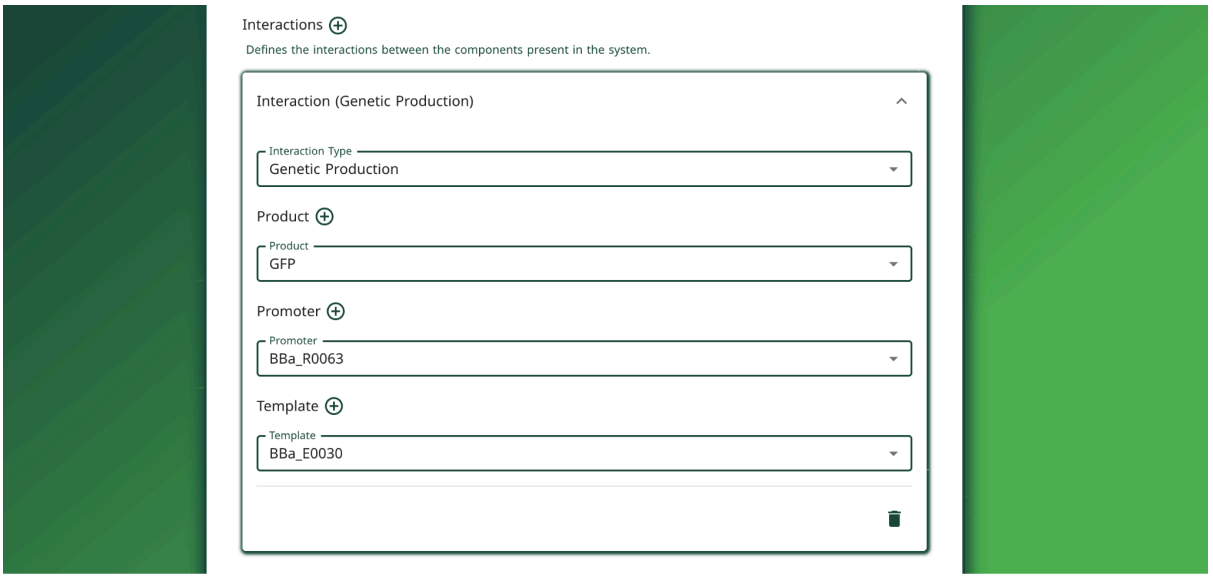
- **Defining Operons:**
  - Select the components that will form each operon.
  - Specify the connections between components (e.g., "Order" or "Regulatory" types).



- **External Components:** Represent external substances interacting with the system. Specify:
  - URI (Optional)
  - Type (Protein, Simple Chemical, Restriction Enzyme)
  - Name



- **Interaction Definitions:** Specify the type of interaction and select the components involved.



Step 5: File Generation and Conversion

SBOL3.1 File

- Once the configuration is complete, click the "Create SBOL3.1 File" button to generate the document.

Format Conversion (Optional)

- After generating the SBOL3.1 file, the tool enables conversion to FASTA and GenBank formats.
- Users can select both formats, one, or none, depending on their needs.