

MINERVA Home

User manual

Admin manual

Examples

MINERVA configuration: Examples

Table of contents

- [CellDesigner file upload and annotation](#)
 - [Automatic annotation](#)
 - [Automatic verification](#)
 - [Procedural semantic zoom](#)
- [Custom data upload by registered users](#)
 - [Basic format](#)
 - [Advanced format](#)
 - [By-identifier coloring](#)
 - [Custom colors and interaction coloring](#)
- [Advanced file upload](#)

CellDesigner file upload and annotation

This section provides examples of usage of automatic annotators, automatic verifiers and construction of procedural semantic zoom.

This is [Example File 1](#), containing a CellDesigner file. The file is quite simple, and has the following properties:

- Annotations
 - Gene component (SNCA) is not annotated
 - RNA component (SNCA) is annotated with Entrez Gene id (NCBI Gene in CellDesigner MIRIAM tab)
 - one protein component (SNCA) is annotated with UniProt ID, while the other (PARK2) is not annotated
 - Phenotype component (UPS) is annotated with Gene Ontology
 - one interaction **re3** is annotated with PubMed id, while others are not
- Compartments and visual layers
 - **Translation** and **Degradation** visual layers are drawn on top of the diagram
 - **cell** and **nucleus** compartments are drawn

Automatic annotation

See [Admin manual - Configure automatic annotation](#) for details on annotators.

1. Upload the **Example File 1** to your MINERVA instance using [Add project button](#)
2. Configure automatic annotations by:
 1. checking the checkbox **Annotate model automatically**
 2. pressing the **Advanced** button; set the following:
 1. for **Element > Species > Gene**, set HGNC annotator
 2. for **Element > Species > Phenotype**, set Gene Ontology annotator
 3. for **Element > Species > Rna**, set Entrez annotator
 4. for **Element > Species > Protein > GenericProtein**, set Uniprot annotator
3. Press **Submit**

Examine the generated content. Please note that:

- Annotations between SNCA DNA, RNA and Protein are different, because different annotators used different information sources to obtain the information on the elements

- Even though SNCA DNA had no CellDesigner annotation, it was correctly annotated by name by HGNC annotator
- Uniprot annotator works only by identifier, so it was unable to provide annotations for PARK2
- Gene Ontology annotator retrieved a definition of the annotated Phenotype

Automatic verification

See Section [Configure automatic verification](#) for details on annotators.

NOTE: Your MINERVA instance remembers the settings from the previous run, so keep annotation settings as above.

1. Upload the **Example File 1** to your MINERVA instance again. Make sure to either remove previously uploaded example, or to name the uploaded project differently (use [Add project button](#) button)
2. Configure automatic verification by:
 1. checking the checkbox **Verify manual annotations**
 2. pressing the **Advanced** button; set the following:
 1. for **Element > Species > Gene**
 - set HGNC, HGNC symbol and Entrez Gene as *Valid*
 - check *Require annotations* checkbox and as *Required* set HGNC, HGNC symbol and Entrez Gene
 2. for **Element > Species > Phenotype**, set Gene Ontology annotator
 - set Gene Ontology as *Valid*
 - check *Require annotations* checkbox and as *Required* set Gene Ontology
 3. for **Element > Species > Rna**
 - set HGNC, HGNC symbol and Entrez Gene as *Valid*
 - check *Require annotations* checkbox and as *Required* set HGNC, HGNC symbol and Entrez Gene
 4. for **Element > Species > Protein > GenericProtein**
 - set Uniprot as *Valid*
 - check *Require annotations* checkbox and as *Required* set Uniprot
 5. for **Reaction > StateTransitionReaction**
 - set Pubmed as *Valid*
 - check *Require annotations* checkbox and as *Required* set Pubmed
 6. for **Reaction > TranslationReaction**
 - set Pubmed as *Valid*
 - uncheck *Require annotations* checkbox
 7. for **Reaction > TranscriptionReaction**
 - set Pubmed as *Valid*
 - uncheck *Require annotations* checkbox
 8. Make sure no items in upper hierarchy levels (e.g. Element, Species or Protein for GenericProtein) have different sets of verification rules
3. Press **Submit**

Upon upload, an exclamation mark icon will appear next to the project. After clicking, the popup window will look like this:

WARNINGS

No	Content
1	[Gene s2] contains invalid annotations: Ensembl(ENSG00000145335),
2	[Gene s2] contains invalid annotations: RefSeq(NM_000345),
3	[Rna s3] contains invalid annotations: Ensembl(ENSG00000145335),
4	[GenericProtein s9] misses one of the following annotations: Uniprot,
5	[StateTransitionReaction re4] misses one of the following annotations: PubMed,

Download

Examine the generated warnings. Please note that:

- Automatic annotators provided additional annotations to Gene and RNA elements, outside of the **Valid** list, you can suppress these warnings by updating the **Valid** list
- GenericProtein element (Parkin) was indicated as missing a required annotation (Uniprot)
- **StateTransitionReaction** re4 was indicated as missing a required annotation (Pubmed)
- **TranslationReaction** and **TranscriptionReaction** did not raise any warnings as no annotation was required for them

NOTE: Valid and Required settings for lower hierarchy levels are overridden by settings for upper hierarchy levels. For instance, if no PubMed annotation is required for TranslationReaction, but Reaction requires PubMed, reactions re1 and re2 raise a warning.

Procedural semantic zoom

[Example File 1](#) after upload and submit will generate a view with procedurally-generated semantic zoom. Please, note that on the highest zoom out level only one compartment, **cell**, is visible. After zoom-in, two gray areas become visible, describing functional areas in the map. After another zoom-in, they are revealed and the **nucleus** compartment becomes visible. After a final zoom-in, **nucleus** becomes transparent as well. See also [Add project button](#).

Custom data upload by registered users

This section provides examples of data upload by registered users. Two subsection provide detailed examples for uploading basic and advanced formats.

Basic format

The file in basic format has two tab-separated columns **Name** and **Value** (see also [Upload custom data - format](#)):

- **Name** containing names of elements to be colored
- **Value** containing a value from [-1,1] range, to be transformed to red-green coloring, with negative values colored red, positive colored green; the saturation of the color will be defined by the absolute value.

This is [Example File 2](#), containing a custom coloring for an example map above ([Example File 1](#)).

To upload this data set, do the following:

1. Make sure you have the privileges to generate layouts in the example map
 1. Click on **Project detail** (magnifying glass icon in the rightmost column)
 2. See that by your name the **View project** checkbox is checked
2. Enter the **User view** of the example map: from the **Admin view > Map viewer**, click on the name of the uploaded project (**example_map** by default)
3. Make sure that you are logged in by checking the **Profile** tab in the left panel
4. In **Layouts > Custom layouts** tab, use **+Choose** button to select **Example File 2** and press generate
5. In **Layouts > Custom layouts** tab, use the **View** icon (magnifying glass) to examine generated view

Examine the generated coloring and the uploaded data set. Please note that:

- All SNCA elements are colored to the same color because of by-name matching
- SNCA elements are red-colored (the value in the file is -0.25), while PARK2 is green (the value in the file is 0.75)

Advanced format

Advanced format foresees two parts of the uploaded dataset - header and body (see also [Upload custom data - format](#)).

Header lines have to start with '#'. It can contain the following elements:

- **Version** # `VERSION=xyz` - a version of this custom layout
- **Name** # `NAME=xyz` - a name that will be automatically assigned upon upload
- **Description** # `DESCRIPTION=xyz` - a description that will be automatically assigned upon upload

Body is a table with a following set of columns:

- **Name, Value** - same as in basic layout
- **Compartment** - name of a compartment in which coloring should take place
- **Chebi** - ChEBI identifiers of elements to be colored
- **Entrez gene** - Entrez identifiers of elements to be colored
- **Gene ontology** - Gene Ontology identifiers of elements to be colored
- **Ensembl** - Ensembl identifiers of elements to be colored
- **Hgnc symbol** - HGNC symbols of elements to be colored
- **Uniprot** - Unprot identifiers of elements to be colored
- **ReactionIdentifier** - ID of interaction to be colored (interactions coloring only)
- **LineWidth** - linewidth of the colored interaction (interactions coloring only)
- **Color** - color of the colored element.

By-identifier coloring

This is [Example File 3](#), containing a custom coloring for an example map above ([Example File 1](#)).

Upload this data set similarly as in [basic format upload](#).

Examine the generated coloring and the uploaded data set. Please note that:

- elements are colored by their identifiers instead of their names
- groups of SNCA elements are colored differently - nucleus elements are colored red, cell elements are colored green; this is a combination of two factors
 - by compartment constraint (red SNCA elements in **nucleus**)
 - by identifier constraining (SNCA elements in **cell** compartment only are annotated with uniprot)
- removing **compartment** column will cause an error, because due to automatic annotation SNCA elements in **cell** compartment are also annotated with **Entrez gene** identifier

- using SNCA coloring by-name overrides by-identifier coloring, the order of selection is as follows:
 - match by-name
 - match by-identifier
 - constrain by-compartment

Custom colors and interaction coloring

Advanced format foresees assigning user-defined colors to both elements and interactions.

This is [Example File 4](#), containing a custom coloring for an example map above ([Example File 1](#)).

Upload this data set similarly as in [basic format upload](#).

Examine the generated coloring and the uploaded data set. Please note that:

- **color** column replaces **value** for defining colors of interactions and elements
- **reactionIdentifier** is not affected by **compartment** constraint

Advanced file upload

This is [Example File 5](#), containing a .zip archive with the following structure:

- **example_advanced_map.xml** - CellDesigner file with the main map
- **data_mining** - folder containing a text file **data_mining_test.txt**
- **images** - folder containing:
 - image files **overview_main.png** and **overview_sub.png**
 - **coords.txt** file containing relationships between the images and the content of the map
- **layouts** - folder containing a text file **predefined_layout.txt**
- **submaps** - folder containing:
 - a submap CellDesigner file **example_submap.xml**
 - a mapping CellDesigner file **mapping.xml**

1. Upload the **Example File 1** to your MINERVA instance using [Add project button](#)
2. Examine the upload window, note the automatically filled positions
3. Make sure that in the **Complex** section proper checkboxes are checked for the main map and the mapping file \$. In the **Complex** section, you can set the names of uploaded submaps

Examine the generated project. Please note that:

- **Data mining** files: candidate interactions have been uploaded for SNCA elements, but not for PARK2; the reason is the lack of annotation - configuration of automatic annotators does not provide PARK2 with MIRIAM identifier, and the linking to file in **data_mining** folder is not successful
- **Images** files
 - **Show overview** button is displayed, invoking one of the uploaded images
 - Links between the images and links directly to the map are configured
 - The link from *Active area 3* invokes a search query, highlighting two interactions
- **Layout** files one predefined layout **Advanced layout** is available for non-registered users
- **Submap** files
 - in the **Submap** tab, the uploaded **example_submap** is visible
 - in the **Search** tab, if you look for PARK2 or SNCA, the results in both main map and the submap will be shown
 - contents of the **example_submap** are not annotated, as the **example_submap.xml** file elements are not annotated with relevant MIRIAM identifiers; even though SNCA elements are annotated in the main map, SNCA element in the submap is not annotated.
 - upon switching to a custom coloring (**Layouts tab > Advanced Layout**) contents of the **example_submap** are colored as the contents of the main map