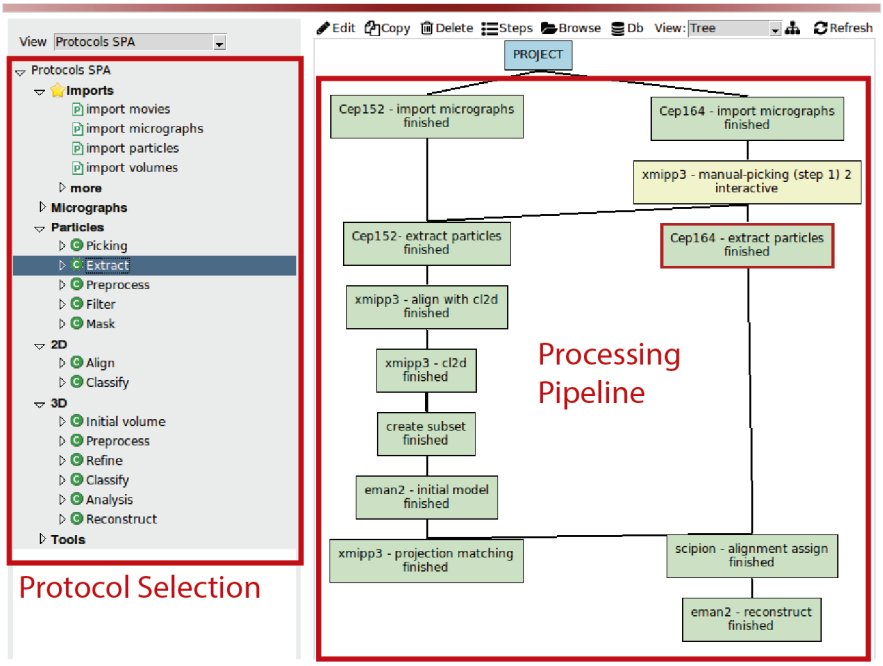
Single particle reconstruction from SMLM images in Scipion

Single-particle reconstructions were implemented using the software package Scipion[11]. We decided to use Scipion for multiple reasons: 1) it is free, 2) it integrates many widely-used EM reconstruction routines such as Xmipp[12], Relion[13], Spider[14] or Eman[15], 3) it has a structured, reproducible workflow and 4) it includes functionalities critical for multi volume alignment. The software is available for Linux operating systems and was used here on a Windows 10 computer running Ubuntu 16.04 LTS in a virtual machine (VMware Workstation).

Scipion can be downloaded as a universal binary (https://github.com/I2PC/scipion/wiki/How-to-Install-1.1-bin), which only needs to be configured and does not require further installation. Available EM reconstruction packages can be added or removed later.

Scipion Main Window



After starting the software and creating a new project, the Scipion main window opens. The program shows an overview of all the integrated software protocols, classified into applications on the left. The main part of the window shows the different processing steps as a flow diagram. Shown is the basic workflow for 2C volume reconstruction using Cep152 (left) as a reference. The workflow for Cep152 (left column) also resembles the workflow for single color reconstruction.

The following steps describe the general reconstruction pipeline:

1) Import Images

The generated image galleries, as for example those shown in Fig. S4 and S7, are imported into Scipion. Example stitched particle montages are provided in the *data* folder of the **Supplementary Software** and on GitHub (https://github.com/christian-7/MultiColorSPR). Within the software, the montage images are generated within the script *particle_filter.m* (see also **Supplementary Note 4**).

2) Pick and Extract particles

During the next step, the individual particles need to be identified (i.e. picked) and extracted from the imported images. For this step, we used the protocol *Manual Picking (Xmipp3)*. This module combines manual and automated identification of particles after selecting a minimum of 15 of them as a training data set. Due to the prior filtering, noise reduction and alignment of the extracted particles, this step was in general very fast and the automated particle picker could easily identify the rendered particle. The picked particles were then extracted using *Extract particles (Xmipp3).* For two-color datasets, we only picked one channel and then extracted both particle sets using the same coordinates.

3) Filtering (optional)

Following particle extraction, the particles can be further filtered or masked. While in some occasions we saw a positive effect after masking the particle with a simple circular mask, this was in general not necessary as the filtering was optimized during the segmentation/filtering step in MATLAB.

4) Align

The particles were aligned using *Cl2D* (*Xmipp3*) without a reference image and a maximum shift of 20 pixel.

5) Classification

The aligned particles were classified in order to generate class averages that can then be used for 3D volumetric reconstruction. The number of classes needs to be set by the user and essentially depends on the symmetry of the structure as each class average resembles one orientation. For the centriole, we typically used 10-15 classes and tried to populate each class with a minimum of 100 particles. Within Scipion, the particles were classified using either *ML2D* or *Cl2D* (both *Xmipp3*). Both classification routines are template-free and provided comparable results on our data sets. We then visually inspected the classification result and selected the class averages that 1) best resembled the particles that went into the class and 2) still corresponded to the expected shape (in different orientations) of the structure as visualized using STORM. Typically, 1-2 classes could be removed as the software classified insufficiently or

inhomogeneously labelled particles not suitable for further processing. This step is therefore critical not only to identify particle orientations, but also as an additional filtering step.

6) Generate initial model

The selected class averages were then used to build an initial model using *initial model* (*Eman2*). Here the user can set a symmetry parameter if known for the investigated structure. For the centriole, we chose either rotational (c36), 9-fold (c9) or no symmetry. Depending on the investigated protein, we assumed a rotational symmetry after 2D particle averaging and visualization of a ring-like structure. For Cep164 and Cep57 we used a 9-fold symmetry, which was indicated by our 2D top view projection analysis (**Supplementary Fig. 8**). For **Fig. 2** and **Supplementary Fig. 8** we did not apply any symmetry constraint.

7) Refine initial model

Eman2 calculates 10 initial models, where the first one usually has the highest chance of being correct. We inspected the initial models and chose the one (typically the first) that resembled the structure as visualized using STORM. The initial model was then refined using the *projection matching (Xmipp3)* protocol using the same symmetry parameters as for the initial model. The remaining parameters were left at default values.

8) Visualization of the final 3D model

The refined 3D model was loaded into UCSF Chimera and modified (color, orientation, etc) to generate the desired output image.

9) Reconstruction of a second co-oriented protein (optional)

The above described steps 1-8 were used to reconstruct a protein from a single-color dataset (e.g. Cep152 in **Fig. 1a**) and the reference protein from a two-color dataset. In order to reconstruct the protein of interest, the angular information from the refinement of the reference protein was applied to the second particle dataset. For this operation, we used the Scipion function *alignment assign.* As a result, each particle in the second channel dataset is now co-oriented to the respective reference particle. The protein of interest can now be directly reconstructed using *reconstruct* (*Relion or Eman*). For this step, we chose the symmetry property as described above and *Fourier* as reconstruction method.