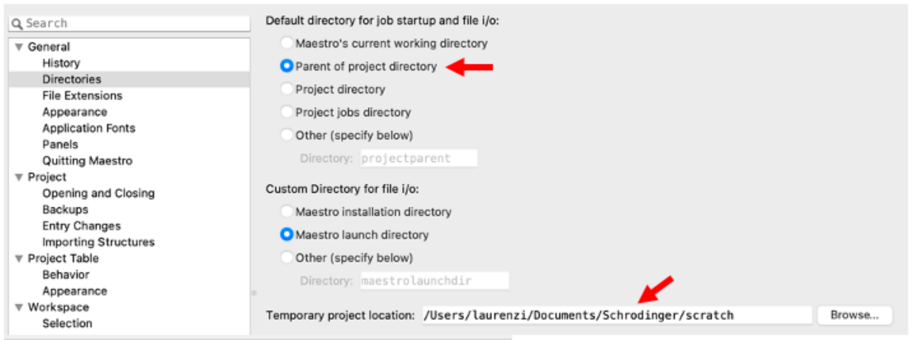
Schrodinger Notes

Good practices

- ***Always know you working directory***. An easy way to always keep track of your working directory is by modifying some settings in the Maestro preferences (under the Maestro menu bar on Mac or File in Linux and Windows).



Default directory for job startup and file i/o:

- Maestro's current working directory
- [✓ Parent of project directory]
- Project directory
- Project jobs directory
- Other (specify below)

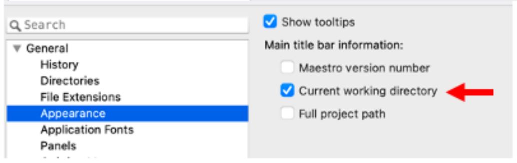
Directory: projectparent

Custom Directory for file i/o:

- Maestro installation directory
- [✓ Maestro launch directory]
- Other (specify below)

Directory: maestrolaunchdir

Temporary project location: /Users/laurenzi/Documents/Schrodinger/scratch



- [✓ Show tooltips]

Main title bar information:

- Maestro version number
- [✓ Current working directory]
- Full project path

- Organize your files! ***Always keep only ONE project per directory.*** In this way, each parent directory will contain one project and its associated I/O files. Note that the project and its parent directory have the same name.

Using the settings above, each time you open a project Maestro will automatically change the working directory to the project's parent directory, that is the directory that contains the project. If you keep only one project per directory, then you can easily organize your files like this:

```
~/Documents/Schrodinger
            |--- my_project_1
            |     |-- job_1
            |     |-- job_2
            |     |-- my_project_1.prj  (<-- the project!)
            |     |-- notes.txt
            |
            |--- my_other_project_2
                  |-- job_1
                  |-- job_2
                  |-- my_other_project_2.prj
                  |-- notes.txt
```

Desmond

Welcome to the dark side of Desmond.

MD Restraints:

- `restraints` keyword (https://www.schrodinger.com/kb/332119)
- `restrain` keyword is described in the Desmond Command Reference in the main Schrodinger documentation

Forcefields:

Useful references:

- https://www.protocols.io/view/how-to-assign-charmm-parameters-to-desmond-generat-bp53mq8n
- https://www.protocols.io/view/how-to-assign-amber-parameters-to-desmond-generate-byyqpxvw (convert antechamber output)
- http://dx.doi.org/10.17504/protocols.io.byyqpxvw
- http://groups.google.com/group/desmond-md-users

Command-line tools:

`$SCHRODINGER/run viparr.py`

Will assign a custom forcefield:

- use `-f FFNAME` to use a built-in ff (listed with `-h`)
- use `-c` to run viparr on selected CT blocks.

`$SCHRODINGER/run build_constraints.py`

This is required to constrain hydrogens. Default settings work fine.

`$SCHRODINGER/run desmond_ff_builder`

Is used to assign the OPLS forcefield. `--keep_ffio_block` is useful if you are mixing forcefields, use this at the end of your workflow. `--skip_fepio` is usually required.

`$SCHRODINGER/run rebuild_cms.py`

If tweaking with CTs, use this with `-make_full_ct` to build the final ”full_system” used for visualization in maestro

System setup

Example WYSIWYG configuration `system.csb` file. Use this with

`$SCHRODINGER/run $SCHRODINGER/utilities/system_builder`.

Keywords are undocumented, but things can be discovered abusing `grep` or `ag` inside schrodinger installation directory.

```
{
  set_oplsaa_version_as OPLS3e

  read_solute_structure "system_setup_6MVD-membrane.mae"

  solvent_desmond_oplsaa_typer {
    input_file_name "spc.box.mae"
    run
  }
  boundary_conditions orthorhombic 85.941 85.941 170.000
  positive_ion_desmond_oplsaa_typer {
    input_file_name "Na.mae"
    run
  }
  negative_ion_desmond_oplsaa_typer {
    input_file_name "Cl.mae"
    run
  }
  neutralize
  salt_positive_ion_desmond_oplsaa_typer {
    input_file_name "Na.mae"
    run
  }
  salt_negative_ion_desmond_oplsaa_typer {
    input_file_name "Cl.mae"
    run
  }
  add_salt 0.150000

  align_principal_axis_with_xyz No
  align_principal_axis_with_diagonal No
  rezero_to_center_of_mass No

  solvate
  write_mae_file "system_setup_6MVD_membrane-out.mae"
}
```

Manually build CTs ...

Enhanced Sampling

Example blocks can be placed in the `.msj` inside the last `simulate` block or anywhere in the `.cfg` file:

Using the `BiasingForce` plugin:

https://www.deshawresearch.com/downloads/download_desmond.cgi/Desmo nd_Users_Guide-0.5.3.pdf#subsection.2.6.2

```
    # BIASINGFORCE
    atom_group = [
        { atom = "asl:a.pt CA"
        index = 1
        name = cm_moi
        }
        { atom = "asl:a.e P and r.pt POPC"
        index = 2
        name = cm_moi
        }
    ]
    backend = {
        force.term = {
            list = [ BiasingForce]
            BiasingForce = {
                type = BiasingForce
                cm_moi = [ {
                    groups = [1 2]
                    distance = 25
                    distance_coeff = 10
                } ]
                output = {
                    first = 0.0
                    interval = 100.0
                    name = "$JOBNAME.bforce"
                }
            }
        }
    }
```

Using enhsamp (The hard way)

- https://www.deshawresearch.com/downloads/download_desmond.cgi/Desmond_Users_Guide-0.5.3.pdf#chapter.9
- https://www.deshawresearch.com/downloads/download_desmond.cgi/Desmond_Users_Guide-0.5.3.pdf#appendix.F

Use the references described above to write an enhsamp script. This example will run a simple targeted MD for one atom.

```
# bias.mexp
declare_output( name     = "bias",
                first    = 0.0,
                interval = 20 );

static p0(3), v(3);
```

```
p_f = array(0, 0, 0); # target positions

gids = atomsel("asl: a.n 1");

sim_time = 100;

k = 0.75;

if time()
    then {
        ref = p0 + v * time();
        p = pos(gids[0]);
        r = norm(ref - p);
        1/2 * k * r^2;
    } else {
        store(p0, pos(gids[0]));
        store(v, (p_f - pos(gids[0])) / sim_time);
        0;
    };
```

Use the following to translate the script in ark syntax

```
$SCHRODINGER/run enhsamp.py bias.mexp > bias.ark
```

Then place the content of `bias.ark` in the `.cms` or `.cfg` files (as described above).

WARNING: `enhsamp.py` is bugged and you should edit *the first few words of its output* so that the final block looks like this:

```
  backend.force.term = {
    list[+]=ES
    ES={type=enhanced_sampling
    gids=[...]
    sexp=[...]
    metadynamics_accumulators=[] storage={...} name="bias" first=0.0 interval=20.0}}
}
```

Indentation is not relevant in any way, but make sure that closing parentheses `{}` are paired correctly.

Change the beginning of the ark output from:

```
force.term{list[+]=ES ES=force.term{list[+]=ES ES={type=enhanced_sampling ...
```

to:

```
backend.force.term = { list[+]=ES ES={type=enhanced_sampling ... }
```