

# gfp\_gaussian\_process

Re-implementation of the python code: [https://github.com/fioriathos/new\\_protein\\_project](https://github.com/fioriathos/new_protein_project)

## Compile

```
cd src; make
```

## Run

```
cd bin
```

```
./gfp_gaussian <infile> [-options] with following options:
```

-p, --parameter_config	file defining the type, step, bounds of the parameters
-c, --csv_config	file that sets the columns that will be used from the input file
-m, --mode	mode keyword can start with 'm'→minimization or 's'→scan
-l, --print_level	print level >=0
-r, --rel_tol	relative tolerance of minimization
-h, --help	help message

The columns that are taken from the csv file can be set by modifying `csv_config.txt`, although defaults are set.

The parameter space is defined in `parameter_bound.txt`.

## Approx. time

- calculation of likelihood for 1125 cells and 26552 data points in total takes around 0.014 seconds

## Notes

## Libraries

- Minimization: nlopt
  - can be installed via cmake
  - can be statically compiled easily

- Linear algebra: Eigen
  - available via modules

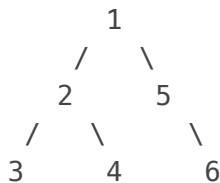
## TODO: Likelihood calculation

☐ check mean and covariance after division

## Likelihood Calculation

- apply function recursively
- every cell is accessed once and after its parent is calculated

```
/* applies the function func to the cell cell and the other cells in the genealogy
 * such that the parent cell has already been accessed when the function is applied
 * to the cell.
 *
 * Example (number implies the order in which)
 * _____
```



```

* _____
*/
void likelihood_recr(const std::vector<double> & params_vec, MOMAdata *cell, double &total_likelihood) {
    /*
     * Recursive implementation that applies the function func to every cell in the genealogy
     * not meant to be called directly, see wrapper below
     */
    if (cell == nullptr)
        return;
    sc_likelihood(params_vec, *cell, total_likelihood);

    likelihood_recr(params_vec, cell->daughter1, total_likelihood);
    likelihood_recr(params_vec, cell->daughter2, total_likelihood);
}

```

## Minimizer

- nlopt

```
void minimize_wrapper(double (*target_func)(const std::vector<double> &x, std::vector<double> &MOMAdata &cell,
Parameter_set &params,
double relative_tol)
```

## Current minimizer: COBYLA

- Constrained Optimization By Linear Approximation (COBYLA)
- Implementation of Powell's method:
  - pick initial  $x_0$  and two directions  $h_1, h_2$
  - starting from  $x_0$  1D optimization along first direction  $h_1 \rightarrow$  find  $x_1$
  - starting from  $x_1$  1D optimization along first direction  $h_2 \rightarrow$  find  $x_2$
  - $h_3$  connects  $x_0$  and  $x_2$
  - starting from  $x_2$  1D optimization along first direction  $h_3 \rightarrow$  find  $x_3$

## Parameter file

with connfig file containing :

- parameter = value, step
- parameter = value, step, lower, upper
- parameter = value
- ...

```
mean_lambda = 0.01, 1e-4
gamma_lambda = 0.01, 1e-4, 1e-4, 0.05
var_lambda = 1e-07
```

## Parameters

- Growth rate fluctuations params:
  - mean\_lambda;
  - gamma\_lambda;
  - var\_lambda;
- gfp fluctuation params
  - mean\_q;
  - gamma\_q;
  - var\_q;
  - beta;
- variance guess for length and gfp

- `var_x;`
  - `var_g;`
- cell division:
  - `var_dx;`
  - `var_dg;`