# C++ Documentation

## Quick-Start

Files you need:

1. Code: *extract\_counts.cpp, extract\_counts.h, extract\_counts.o*
2. FASTA file
3. Structure file: *structure.txt*
4. All the code lists specified in the structure file

Put all the files in the same folder and execute the following commands in the terminal:

|  |
| --- |
| g++ -O3 -std=c++11 -o extract\_counts.o extract\_counts.cpp |
| ./extract\_counts.o (*structure*.txt) |

The part in the brackets, i.e. (structure.txt) is optional (leave out the brackets if you use it) and allows you to specify the structure file on the command line directly. If you do not use this, the program will ask you where to find the structure file. The structure file can have an arbitrary name.

## Sequence design

The DNA tags of a DECL library is composed of primer codes, constant regions and building block codes. Every read starts with a primer code which identifies the selection which is performed. The number of primer codes and building block codes can be arbitrary but computational costs to map the reads to the known codes grows exponentially and is thus not feasible beyond a handful of codes.

The arrangement of constant, primer and building block regions can be arbitrary.

## Executable file

In a first step the executable file has to be compiled. You need the following files for this:

1. extract\_counts.cpp
2. extract\_counts.h
3. extract\_counts.o

Put all the files in the same folder and open the terminal. Navigate to the folder with the files (you can do this with *cd* and dragging the folder into the terminal). Now you can produce the executable by typing:

|  |
| --- |
| g++ -O3 -std=c++11 -o extract\_counts.o extract\_counts.cpp |

You should now see a file *extract\_counts.o* in the folder.

## Data Input

### FASTA file

Suitable data input is in form of a fasta file with alternating info line (starting with >) followed by a sequence read.

Example:

>700523F:272:CD2R4ANXX:6:2212:1104:2142 1:N:0:799

GNAGATGGAGCTTCTGAATTCTGTGTGCTGTGCATGCGAGTCCCATGGCGCAGCTGCCACGTCAA

### Structure file

The structure file specifies the design of the DNA tags. And has the following setup:

Description Example

Path of FASTA file ../1812\_Federica\_R1\_truncated\_long.fasta

Name and path of folder to store results Evaluation/

Start End Identifier path 7 30 C ../const1.txt

37 5 C ../const2.txt

66 86 C ../const3.txt

1 6 S ../PrimerCodes1.txt

87 92 S ../PrimerCodes2.txt

31 36 B ../Elib2-789-DENOVO.txt

58 65 B ../Elib4-426-DENOVO.txt

Start indicates where the specified code starts in the read

End indicates where the specified code ends in the read

Identifier specifies what type of code is at this location, i.e. a constant region, a primer region or a building block region. This information is important to generate the output.

Path indicates where the list of codes at this location can be found.

In brief, to identify the different selection the algorithm will generate an output file for each combination of primer codes (if at least one read is found for this selection). The primer codes used are stored in the corresponding files specified in the path of the S type codes (S = Selection). The same is true for the building block codes and the constant regions.

## Data Output

The program will generate the specified folder in the second line of the *structure* file (i.e. /Evaluation in the example above). For each selection a *txt file* is created that contains the count of each B code combination. If for a selection no read is found the file will be deleted again, i.e. if you cannot find the desired selection in the output folder this particular selection did not have any read.