Sorry, my explanation of the “Python Effector - Rotate and lift” was not complete or detailed enough. Here some additional information and some background.

First of all, MoGraph and thus the MoGraph Cloner is re-calculated on every frame and scene change.

Second MoGraph has several arrays that control everything. Here is an overview:

MODATA_MATRIX	Matrix of the clone.
MODATA_COLOR	Color of the clone.
MODATA_SIZE	Size of the clone.
MODATA_UVW	UV position of the clone.
MODATA_FLAGS	Flags:
MODATA_WEIGHT	Weight of the clone.
MODATA_CLONE	Clone Offset
MODATA_TIME	Time offset of the clone.
MODATA_LASTMAT	Previous frame particle matrix.
MODATA_STARTMAT	Matrix at the particle’s birth.
MODATA_ALT_INDEX	Alternative index
MODATA_FALLOFF_WGT	Falloff weight.

See for more information the Cinema 4d Python documentation.

With our effector we are going to change – only - the MODATA_MATRIX array which holds the position, scale and rotation of the clones. In this case we are only going to change the position of the clones.
Because MoGraph is re-calculated on every frame, we have to set the MODATA_MATRIX with our effector on every frame.

Ok, back to our effector.
Basically it is very simple, initially we copy the MoGraph MODATA_MATRIX to our own matrix array (newMarr) and set (lift) the position for the first index of the new array to +100. Then we shift (rotate) our newMarr array every N frames (here every 15 frames). At the end, before we the effector is ended, we write our newMarr array to the MoGraph Cloner.

The next time the effector is called, we check whether N frames have passed and if so, we shift our newMarr array again. At the end, we write our newMarr array back to the MoGraph Cloner.

Note: Again, we only change the position of the clones, not the order or the index of the clones. You can see this because, for example, the color of the clones is not changed!

Here part of the updated and more simple code:

```
def shiftArray(key, array):
    return array[-key:]+array[:-key]

def main():
    md = mo.GeGetMoData(op)
    if md==None: return False

    cnt = md.GetCount()
    marr = md.GetArray(c4d.MODATA_MATRIX)

    global newMarr
    #initially - only once
    try:
        marr = newMarr
    except NameError:
        newMarr = marr                                   #copy MoGraph array to our own array
        opm = utils.MatrixMove(c4d.Vector(0, 100.0, 0))  #lift first - only once
        newMarr[0] = newMarr[0]*opm

    # shift the array every N (15) frames.
    frame = doc.GetTime().GetFrame(doc.GetFps())
    if frame%15==0:
        newMarr = shiftArray(1, marr)

    #write our array back to the MoGraph array
    md.SetArray(c4d.MODATA_MATRIX, newMarr, False)
    return True
```

To show that MoGraph is re-calculated every frame and thus we have to write back every frame, try the following. Instead of writing back every frame, do it only when our array has changed, so only every N frames. See the code below.

```
    # shift the array every N (15) frames.
    frame = doc.GetTime().GetFrame(doc.GetFps())
    if frame%15==0:
        newMarr = shiftArray(1, marr)
        md.SetArray(c4d.MODATA_MATRIX, newMarr, False)   #Try!!!

    #write our array back to the MoGraph array
    #md.SetArray(c4d.MODATA_MATRIX, newMarr, False)      #Comment this line out when trying
    return True
```

You can now see that on every N frame the clone is lifted. On other frames we see the standard Clones, without our effector!

Note: Instead of the NameError trick, you can use a global Boolean.