```
OUTPUT_DIR ?= Release

BUILD ?= MPI
#set BUILD to MPI to force MPI, should be set in ../Makefile

MPIF90C ?= mpif90

ifortErr = $(shell which ifort >/dev/null; echo $$?)

#these settings for ifort 14 and higher. Earlier versions will not work.
ifeq "$(ifortErr)" "0"
#ifort; Can remove -xHost if your cluster is not uniform, or specify specific
processor optimizations -x...

F90C     = ifort
#use this if mpif90 is trying to use gfortran: MPIF90C = mpif90 -f90=ifort

FFLAGS = -mkl -O3 -no-prec-div -fpp
DEBUGFLAGS = -mkl -g -check all -check noarg_temp_created -traceback -fpp -fpe0
#add -fpe0 to check for floating point errors (think lowLike also throws these
harmlessly)
MODOUT = -module $(OUTPUT_DIR)
LAPACKL =

ifortVer_major = $(shell ifort -v 2>&1 | cut -d " " -f 3 | cut -d. -f 1)
ifeq ($(shell test $(ifortVer_major) -gt 15; echo $$?),0)
FFLAGS+= -qopenmp
DEBUGFLAGS+= -qopenmp
else
FFLAGS+= -openmp
DEBUGFLAGS+= -openmp
endif

ifneq ($(NERSC_HOST),)
#setting for Edison/Cori on NERSC
MPIF90C = ftn
FFLAGS += -dynamic
else
ifeq ($(STATIC),YES)
#so you can re-distribute the binary to another machine
#static link details from https://software.intel.com/en-us/articles/intel-mkl-link-
line-advisor
LAPACKL =  -Wl,--start-group $(MKLROOT)/lib/intel64/libmkl_intel_lp64.a $(MKLROOT)/
lib/intel64/libmkl_core.a $(MKLROOT)/lib/intel64/libmkl_intel_thread.a -Wl,--end-
group -lpthread -lm
FFLAGS+=  -axAVX,SEE4.2 -static-intel
else
FFLAGS += -xHost
endif
endif

else
#gfortran 5.x and earlier will not work (many F2003 bugs)
#gfortran 6 should work
#see CosmoBox virtual machines for stable environments
F90C = gfortran
FFLAGS = -cpp -O3 -ffast-math -ffree-line-length-none -fopenmp -fmax-errors=4
DEBUGFLAGS = -cpp -g -fbounds-check -fbacktrace -ffree-line-length-none -fopenmp -
fmax-errors=4 -ffpe-trap=invalid,overflow,zero
MODOUT =  -J$(OUTPUT_DIR)
#LAPACKL = -Wl,-framework -Wl,accelerate
```

```
#commented above is (I think) for Mac; this is standard linux (sudo apt-get install
liblapack-dev)
LAPACKL = -lblas -llapack

#march native does not work on Mac, otherwise use (comment out if your cluster is
inhomogeneous)
ifneq ($(shell uname -s),Darwin)
#native optimization does not work on Mac
FFLAGS+=-march=native
endif

endif

#use "make RECOMBINATION=cosmorec" to build with CosmoRec rather than RECFAST default
RECOMBINATION ?=recfast

EXTDATA=

#use PRECISION=SINGLE to use single precision (probably doesn't work)
PRECISION ?=

#set WMAP empty not to compile with WMAP, e.g. WMAP = /scratch/../WMAP9/likelihood_v5
WMAP ?= /home/ealmaraz/software/wmap/wmap_likelihood_v5

#e.g. PICO=/path/to/pypico
PICO ?= $(shell python -c "import pypico; print pypico.get_folder()" 2> /dev/null)

#Needed for WMAP; set from ~/.bashrc definition or edit here
cfitsio ?= /home/ealmaraz/software/cfitsio/cfitsio3370

#not used in public cosmomc
NONCLIKLIKE ?=

IFLAG = -I
INCLUDE =

#clik path and library
ifneq ($(CLIK_PATH),)
CLIKL = -L$(CLIK_PATH)/lib -lclik_f90
INCLUDE = $(IFLAG)$(CLIK_PATH)/include
endif

ifeq ($(OUTPUT_DIR),Debug)
FFLAGS=$(DEBUGFLAGS)
endif


ifeq ($(BUILD),MPI)
override OUTPUT_DIR :=$(OUTPUT_DIR)MPI
FFLAGS +=  -DMPI
LAPACKL +=  $(LINKMPI)
F90C = $(MPIF90C)
endif

#would like to embed this somehow..
#GIT_HASHTAG = "git log --pretty=format:'%H:%cd' -n 1"

CALCFILES = $(OUTPUT_DIR)/Calculator_CAMB.o

#Can use params_H if you prefer more generic parameters
PARAMETERIZATION = $(OUTPUT_DIR)/CosmologyParameterizations.o
```

```
GETDISTFLAGS = $(FFLAGS)
F90FLAGS = $(FFLAGS) $(IFLAG)../camb/$(OUTPUT_DIR) $(INCLUDE)
LINKFLAGS = -L../camb/$(OUTPUT_DIR) -lcamb_$(RECOMBINATION) $(LAPACKL) $(F90CRLINK)
$(CLIKL)

ifneq ($(PICO),)
ifneq ($(PICO),NO)
CALCFILES += $(OUTPUT_DIR)/fpico_interface.o $(OUTPUT_DIR)/Calculator_PICO.o
LINKFLAGS +=  $(shell python -c "import pypico; print pypico.get_link()")
F90FLAGS += -DPICO $(shell python -c "import pypico; print pypico.get_include()")
endif
endif


UTILS = $(OUTPUT_DIR)/MiscUtils.o $(OUTPUT_DIR)/ArrayUtils.o $(OUTPUT_DIR)/
StringUtils.o $(OUTPUT_DIR)/MpiUtils.o $(OUTPUT_DIR)/FileUtils.o \
    $(OUTPUT_DIR)/RandUtils.o $(OUTPUT_DIR)/ObjectLists.o $(OUTPUT_DIR)/
Interpolation.o

DISTFILES = $(UTILS) $(OUTPUT_DIR)/IniObjects.o \
    $(OUTPUT_DIR)/ObjectParamNames.o $(OUTPUT_DIR)/Matrix_utils_new.o \
    $(OUTPUT_DIR)/settings.o $(OUTPUT_DIR)/samples.o $(OUTPUT_DIR)/IO.o $(OUTPUT_DIR)/
GetDist.o


ifneq ($(PRECISION),)
FFLAGS += -D$(PRECISION) -DMATRIX_$(PRECISION)
endif

PLANCKLIKEFILES= $(OUTPUT_DIR)/CMB_BK_Planck.o

ifneq ($(CLIK_PATH),)
FFLAGS += -DCLIK
PLANCKLIKEFILES += $(OUTPUT_DIR)/cliklike.o
endif

ifeq ($(NONCLIKLIKE),cliklike_CamSpec)
FFLAGS += -DNONCLIK
PLANCKLIKEFILES +=  $(OUTPUT_DIR)/temp_like.o
PLANCKLIKEFILES += $(OUTPUT_DIR)/cliklike_CamSpec.o
endif

LIKEFILES =  $(OUTPUT_DIR)/DataLikelihoods.o $(OUTPUT_DIR)/calclike.o

OBJFILES = $(UTILS) $(OUTPUT_DIR)/IniObjects.o $(OUTPUT_DIR)/ObjectParamNames.o \
        $(OUTPUT_DIR)/Matrix_utils_new.o  $(OUTPUT_DIR)/settings.o \
        $(OUTPUT_DIR)/samples.o $(OUTPUT_DIR)/IO.o $(OUTPUT_DIR)/likelihood.o
$(OUTPUT_DIR)/GeneralTypes.o $(OUTPUT_DIR)/BaseParameters.o \
        $(OUTPUT_DIR)/propose.o $(OUTPUT_DIR)/ParamSet.o $(OUTPUT_DIR)/
PowellConstrainedMinimize.o $(OUTPUT_DIR)/ImportanceSampling.o \
        $(OUTPUT_DIR)/MCMC.o $(OUTPUT_DIR)/minimize.o $(OUTPUT_DIR)/SampleCollector.o
$(OUTPUT_DIR)/GeneralSetup.o


SUPERNOVAE = $(OUTPUT_DIR)/supernovae_Union2.o $(OUTPUT_DIR)/supernovae_SNLS.o
$(OUTPUT_DIR)/supernovae_JLA.o
SZ = $(OUTPUT_DIR)/szcounts.o

DATAMODULES = $(PLANCKLIKEFILES) $(OUTPUT_DIR)/mpk.o $(OUTPUT_DIR)/wigglez.o \
        $(OUTPUT_DIR)/bao.o $(SUPERNOVAE) $(SZ) $(OUTPUT_DIR)/supernovae.o
$(OUTPUT_DIR)/HST.o  $(OUTPUT_DIR)/CMB.o $(OUTPUT_DIR)/CMBlikes.o $(OUTPUT_DIR)/
ElementAbundances.o
```

```
DATAMODULES += $(OUTPUT_DIR)/wl.o

COSMOFILES = $(OUTPUT_DIR)/CosmologyTypes.o $(OUTPUT_DIR)/CosmoTheory.o $(OUTPUT_DIR)/
bbn.o $(OUTPUT_DIR)/Calculator_Cosmology.o \
        $(OUTPUT_DIR)/CalcLike_Cosmology.o $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(CALCFILES) $(DATAMODULES) $(PARAMETERIZATION) $(LIKEFILES) \
        $(OUTPUT_DIR)/CosmologyConfig.o

OBJFILES += $(COSMOFILES) $(OUTPUT_DIR)/driver.o

F90CRLINK =

ifeq ($(RECOMBINATION),cosmorec)
## This is flag is passed to the Fortran compiler allowing it to link C++ (uncomment
the right one).
# GCC (gfortran/g++)
COSMOREC_PATH ?= ../CosmoRec/
F90CRLINK = -L$(COSMOREC_PATH) -lCosmoRec -L$(GSLPATH)/lib -lgsl -lgslcblas -lstdc++
# Intel Compilers (ifort/icpc)
#F90CRLINK = -cxxlib -L$(COSMOREC_PATH) -lCosmoRec -L$(GSLPATH)/lib -lgsl -lgslcblas
FFLAGS +=  -DCOSMOREC
endif

ifeq ($(RECOMBINATION),hyrec)
HYREC_PATH ?= ../HyRec/
F90CRLINK += -L$(HYREC_PATH) -lhyrec
endif

default: cosmomc

$(OUTPUT_DIR)/StringUtils.o: $(OUTPUT_DIR)/MiscUtils.o
$(OUTPUT_DIR)/RandUtils.o: $(OUTPUT_DIR)/MpiUtils.o
$(OUTPUT_DIR)/FileUtils.o: $(OUTPUT_DIR)/MpiUtils.o $(OUTPUT_DIR)/MiscUtils.o
$(OUTPUT_DIR)/StringUtils.o
$(OUTPUT_DIR)/ObjectLists.o: $(OUTPUT_DIR)/FileUtils.o
$(OUTPUT_DIR)/IniObjects.o: $(OUTPUT_DIR)/FileUtils.o
$(OUTPUT_DIR)/Interpolation.o: $(OUTPUT_DIR)/ObjectLists.o
$(OUTPUT_DIR)/Matrix_utils_new.o: $(OUTPUT_DIR)/FileUtils.o
$(OUTPUT_DIR)/ObjectParamNames.o: $(UTILS) $(OUTPUT_DIR)/ObjectLists.o
$(OUTPUT_DIR)/settings.o:  $(OUTPUT_DIR)/IniObjects.o $(OUTPUT_DIR)/ObjectParamNames.o
$(OUTPUT_DIR)/propose.o: $(OUTPUT_DIR)/settings.o $(OUTPUT_DIR)/Matrix_utils_new.o
$(OUTPUT_DIR)/GeneralTypes.o
$(OUTPUT_DIR)/IO.o: $(OUTPUT_DIR)/settings.o
$(OUTPUT_DIR)/samples.o: $(OUTPUT_DIR)/ObjectLists.o $(OUTPUT_DIR)/settings.o
$(OUTPUT_DIR)/Matrix_utils_new.o
$(OUTPUT_DIR)/GetDist.o: $(OUTPUT_DIR)/IO.o $(OUTPUT_DIR)/samples.o

$(OUTPUT_DIR)/CalcLike_Cosmology.o: $(OUTPUT_DIR)/calclike.o


$(OUTPUT_DIR)/likelihood.o: $(OUTPUT_DIR)/GeneralTypes.o
$(OUTPUT_DIR)/GeneralTypes.o: $(OUTPUT_DIR)/ObjectLists.o $(OUTPUT_DIR)/settings.o
$(OUTPUT_DIR)/IO.o
$(OUTPUT_DIR)/BaseParameters.o: $(OUTPUT_DIR)/GeneralTypes.o $(OUTPUT_DIR)/IO.o
$(OUTPUT_DIR)/settings.o
$(OUTPUT_DIR)/ParamSet.o: $(OUTPUT_DIR)/BaseParameters.o $(OUTPUT_DIR)/samples.o
$(OUTPUT_DIR)/DataLikelihoods.o: $(OUTPUT_DIR)/likelihood.o $(OUTPUT_DIR)/ParamSet.o
$(DATAMODULES)
$(OUTPUT_DIR)/calclike.o: $(OUTPUT_DIR)/DataLikelihoods.o
$(OUTPUT_DIR)/ImportanceSampling.o: $(OUTPUT_DIR)/calclike.o $(OUTPUT_DIR)/IO.o
$(OUTPUT_DIR)/MCMC.o: $(OUTPUT_DIR)/calclike.o $(OUTPUT_DIR)/RandUtils.o
```

```
$(OUTPUT_DIR)/propose.o $(OUTPUT_DIR)/ParamSet.o
$(OUTPUT_DIR)/minimize.o: $(OUTPUT_DIR)/PowellConstrainedMinimize.o $(OUTPUT_DIR)/
MCMC.o
$(OUTPUT_DIR)/SampleCollector.o: $(OUTPUT_DIR)/MCMC.o
$(OUTPUT_DIR)/GeneralSetup.o: $(OUTPUT_DIR)/SampleCollector.o $(OUTPUT_DIR)/
ImportanceSampling.o  $(OUTPUT_DIR)/minimize.o

$(OUTPUT_DIR)/driver.o: $(OUTPUT_DIR)/GeneralSetup.o $(OUTPUT_DIR)/CosmologyConfig.o

$(PARAMETERIZATION): $(OUTPUT_DIR)/ParamSet.o $(OUTPUT_DIR)/bbn.o  $(OUTPUT_DIR)/
Calculator_Cosmology.o
$(OUTPUT_DIR)/Calculator_Cosmology.o: $(OUTPUT_DIR)/CosmoTheory.o
$(OUTPUT_DIR)/CosmoTheory.o: $(OUTPUT_DIR)/CosmologyTypes.o $(OUTPUT_DIR)/likelihood.o
$(OUTPUT_DIR)/supernovae.o:  $(OUTPUT_DIR)/Likelihood_Cosmology.o $(SUPERNOVAE)
$(OUTPUT_DIR)/Calculator_CAMB.o: $(OUTPUT_DIR)/Calculator_Cosmology.o camb
$(OUTPUT_DIR)/CosmologyTypes.o: $(OUTPUT_DIR)/settings.o $(OUTPUT_DIR)/likelihood.o
$(OUTPUT_DIR)/GeneralTypes.o
$(OUTPUT_DIR)/Likelihood_Cosmology.o: $(OUTPUT_DIR)/Calculator_Cosmology.o
$(OUTPUT_DIR)/CosmoTheory.o
$(OUTPUT_DIR)/CMBlikes.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/CMB_BK_Planck.o: $(OUTPUT_DIR)/CMBlikes.o
$(OUTPUT_DIR)/CMB.o: $(OUTPUT_DIR)/CMBlikes.o $(PLANCKLIKEFILES)
$(OUTPUT_DIR)/bbn.o: $(OUTPUT_DIR)/settings.o $(OUTPUT_DIR)/likelihood.o
$(OUTPUT_DIR)/Interpolation.o
$(OUTPUT_DIR)/mpk.o: $(OUTPUT_DIR)/CosmoTheory.o $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/wigglez.o: $(OUTPUT_DIR)/mpk.o
$(OUTPUT_DIR)/bao.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/wl.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/HST.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/cliklike.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/CosmologyConfig.o: $(PARAMETERIZATION) $(OUTPUT_DIR)/GeneralSetup.o
$(OUTPUT_DIR)/Likelihood_Cosmology.o

$(OUTPUT_DIR)/supernovae_Union2.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/supernovae_SNLS.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o
$(OUTPUT_DIR)/supernovae_JLA.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o

$(OUTPUT_DIR)/cliklike_CamSpec.o: $(OUTPUT_DIR)/temp_like.o

$(OUTPUT_DIR)/szcounts.o: $(OUTPUT_DIR)/Likelihood_Cosmology.o

$(OUTPUT_DIR)/Calculator_PICO.o: $(OUTPUT_DIR)/Calculator_CAMB.o $(OUTPUT_DIR)/
fpico_interface.o

$(OUTPUT_DIR)/ElementAbundances.o: $(OUTPUT_DIR)/bbn.o $(OUTPUT_DIR)/
Likelihood_Cosmology.o

ifneq ($(highL),)
$(OUTPUT_DIR)/cliklike_CamSpec.o: $(highL)/libhigh.a
endif


ifneq ($(WMAP),)
$(OUTPUT_DIR)/CMB.o: $(WMAP)/libwmap9.a
F90FLAGS += $(IFLAG)$(cfitsio)/include $(IFLAG)$(WMAP) -DWMAP
LINKFLAGS +=  -L$(cfitsio)/lib -L$(WMAP) -lcfitsio -lwmap9
endif

F90FLAGS += $(MODOUT) $(IFLAG)$(OUTPUT_DIR)/

export FFLAGS
export F90C
```

```
export OUTPUT_DIR
export IFLAG
export MODOUT

directories:
	mkdir -p $(OUTPUT_DIR)

$(OUTPUT_DIR)/%.o: %.c
	$(CC) $(GSLINC) -c $*.c -o $(OUTPUT_DIR)/$*.o

$(OUTPUT_DIR)/%.o: %.f90 Makefile
	$(F90C) $(F90FLAGS) -c $*.f90 -o $(OUTPUT_DIR)/$*.o

$(OUTPUT_DIR)/%.o: %.F90 Makefile
	$(F90C) $(F90FLAGS) -c $*.F90 -o $(OUTPUT_DIR)/$*.o

$(OUTPUT_DIR)/fpico_interface.o: $(PICO)/fpico_interface.f90 Makefile
	$(F90C) $(F90FLAGS) -c $(PICO)/fpico_interface.f90 -o $(OUTPUT_DIR)/
fpico_interface.o

cosmomc: directories camb $(OBJFILES)
	$(F90C) -o ../cosmomc $(OBJFILES) $(LINKFLAGS) $(F90FLAGS)

cosmomc_debug: directories camb $(OBJFILES)
	$(F90C) -o ../cosmomc_debug $(OBJFILES) $(LINKFLAGS) $(F90FLAGS)


clean: cleancosmomc
	cd ../camb; make clean

cleancosmomc:
	rm -f $(OUTPUT_DIR)/*.o $(OUTPUT_DIR)/*.mod ../core
	rm -rf Release*
	rm -rf Debug*

getdist: directories $(DISTFILES)
	$(F90C) -o ../getdist $(DISTFILES) $(LAPACKL) $(GETDISTFLAGS)

camb:
	cd ../camb && \
	$(MAKE) --file=Makefile_main libcamb OUTPUT_DIR=$(OUTPUT_DIR) \
	RECOMBINATION=$(RECOMBINATION) EQUATIONS=equations_ppf NONLINEAR=halofit_ppf

$(highL)/libhigh.a:
	cd $(highL); make libhigh.a;

$(WMAP)/libwmap9.a:
	cd $(WMAP); make libwmap9.a;
```