# Introduction to the CASINO Program

**Neil D. Drummond**

**Department of Physics, Lancaster University**

QMC2015, Kolkata, India

23rd–27th March, 2015

# Overview

- *Development of the CASINO code*
- *Summary of features of CASINO*
- *Interfaces to other electronic-structure packages*
- *Major input and output files*
- *Documentation and other sources of information and support*

# CASINO: History

- Early 1990s: a Fortran 77 development code ("the QMC code") was written by Richard Needs and Guna Rajagopal in Cambridge, assisted by Matthew Foulkes. The code was further developed by Andrew Williamson up to 1995 and by Mike Towler and Paul Kent up to 1998. Different versions were able to treat FCC solids, single atoms and the homogeneous electron gas.

- Late 1990s: it was clear that a modern, general code capable of treating atoms, molecules, polymers, slabs, crystals and model systems was required, not only for the Cambridge group, but for public distribution. Beginning in 1999, a new Fortran 90 code—CASINO—was developed in the group of Richard Needs, initially by Mike Towler, with important contributions from 2002 by Neil Drummond and from 2004 by Pablo López Ríos.

- Many contributers: Andrew Porter, Randy Hood, Andrew Williamson, Dario Alfè, Gavin Brown, Chris Pickard, Rene Gaudoin, Ben Wood, Zoltan Radnai, Andrea Ma, Ryo Maezono, John Trail, Paul Kent, Nick Hine, Alex Badinski, John Jumper, Robert Lee, Norbert Nemec, Lucian Anton, Priyanka Seth, Jonathan Lloyd-Williams, Pascal Bugnion, Elaheh Mostaani, Albert Defusco, Mike Deible, Blazej Jaworowski, . . .

# “CASINO / CASINÒ” (Italian⟶English)

**casino** m. **1** brothel, whorehouse **2** noise . . . **3** mess, ⟨*volg*⟩ cock-up

**casinò** m. casino

# CASINO: Code Design Aims

**Generality** VMC and DMC for systems of arbitrary size and geometry, including atoms, molecules, systems periodic in 1, 2 and 3 dimensions (polymers, slabs and crystalline solids), various electron and electron–hole phases, generalised quantum particles with arbitrary interactions (cold atoms, etc.). Choice of basis sets (plane waves, Gaussians, blips and Slaters) or grids. Interfaces to a wide range of electronic structure codes for generating trial wave functions.

**Portability** Strict Fortran 95. Modern software design. Runs on any parallel (MPI) and serial hardware. Automatic, customisable compilation and setup.

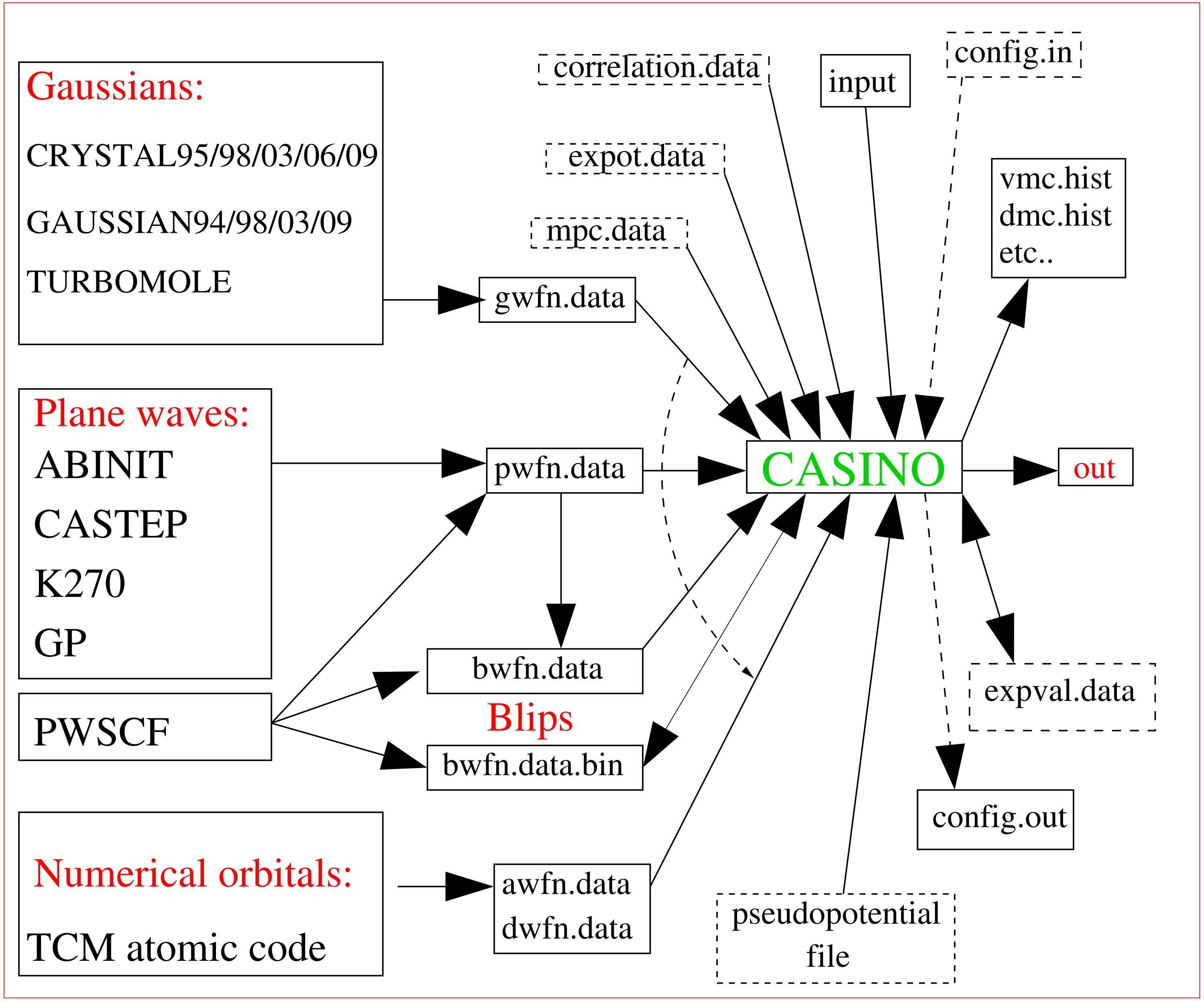
**Ease of use** Shell-script automation. Full documentation: internal help system, comprehensive manual and interactive website, including pseudopotential library: `http://vallico.net/casinoqmc`. Wide range of examples. Discussion forum: `http://vallico.net/tti/casino-forum`.

**Speed and memory efficiency** Efficient algorithms optimised for speed. Best possible scaling with system size and number of processors. Capable of exploiting shared memory (System V, Posix and Blue Gene Posix).

# CASINO: Capabilities

- VMC and DMC.
- Wave-function optimisation by variance or energy minimisation.
- Slater[–Jastrow[–backflow]] trial wave functions. The Slater part may consist of (spin-polarised) multiple determinants. Pairing (geminal) wave functions.
  - Basis functions: plane waves, blips, atom-centred Gaussians ($s$, $sp$, $p$, $d$, $f$ and $g$) with cusp corrections and atom-centred Slater functions. Numerical orbitals for atoms and molecular dimers.
  - Excited states from promotion, addition or subtraction of particles.
- Periodic electron–electron interactions with Ewald, MPC or structure-factor methods.
- Model systems: electron[–hole] gases, positronic molecules, cold atomic gases, . . .
- Expectation values: total energy (and components), charge and spin density, pair correlation function, structure factor, one-particle and two-particle density matrices, electric dipole moment, momentum density, . . .

# CASINO: Interface to Other Packages



Gaussians:

CRYSTAL95/98/03/06/09

GAUSSIAN94/98/03/09

TURBOMOLE

Plane waves:

ABINIT

CASTEP

K270

GP

PWSCF

Numerical orbitals:

TCM atomic code

# CASINO Distribution

```
pyc035000001:~/CASINO> ls
CHANGES  Makefile        VERSION   build-tools/  examples/  manual/
DIARY    README          arch/     data/         install*   src/
FAQ      README_INSTALL  bin_qmc/  doc/          lib/       utils/
```

- Files and directories of particular interest:

  **CHANGES** Summary of major changes to CASINO<br>
  **DIARY** Detailed list of all changes to CASINO<br>
  **FAQ** Frequently asked questions<br>
  **README & README_INSTALL** Introduction & installation instructions<br>
  **VERSION** File containing automatically generated version number<br>
  **install*** Automatic installation/compilation script<br>
  **arch/** Contains architecture-dependent configuration files<br>
  **bin_qmc/** Where executable programs are placed<br>
  **doc/** Contains miscellaneous things to read<br>
  **examples/** Contains large number of examples and test cases<br>
  **manual/** Contains CASINO manual in PDF format<br>
  **src/ & utils/** CASINO source code and source code for utility programs

# Principal Input File: “input” (I)

```
# Silane molecule (ground state)

# SYSTEM
neu               : 4           #*! Number of up electrons (Integer)
ned               : 4           #*! Number of down electrons (Integer)
periodic          : F           #*! Periodic boundary conditions (Boolean)
atom_basis_type   : gaussian    #*! Basis set type (text)

# RUN
runtype           : vmc         #*! Type of calculation (Text)
newrun            : T           #*! New run or continue old (Boolean)

# VMC
vmc_equil_nstep   : 5000        #*! Number of equilibration steps (Integer)
vmc_nstep         : 40000       #*! Number of steps (Integer)
vmc_nblock        : 1           #*! Number of checkpoints (Integer)
vmc_nconfig_write : 0           #*! Number of configs to write (Integer)
```

# Principal Input File: “input” (II)

```
# DMC
dmc_equil_nstep    : 2000            #*! Number of steps (Integer)
dmc_equil_nblock   : 1               #*! Number of checkpoints (Integer)
dmc_stats_nstep    : 10000           #*! Number of steps (Integer)
dmc_stats_nblock   : 1               #*! Number of checkpoints (Integer)
dmc_target_weight  : 1000.0          #*! Total target weight in DMC (Real)
dtdmc              : 0.01            #*! DMC time step (Real)

# OPTIMIZATION
opt_method         : varmin          #*! Opt method (varmin/madmin/emin/...)
opt_jastrow        : T               #*! Optimize Jastrow factor (Boolean)
opt_backflow       : F               #*! Optimize backflow parameters (Boolean)

# GENERAL PARAMETERS
use_jastrow        : T               #*! Use a Jastrow function (Boolean)
backflow           : F               #*! Use backflow corrections (Boolean)

# EXPECTATION VALUES
dipole_moment      : F               #*! Accumulate elec. dipole moment (Boolean)
```

# Principal Input File: “input” (III)

- To discover what an input keyword means, either
  - look it up in the manual **or**
  - use *casinohelp*. For example:

```
pyc035000001:~> casinohelp neu
 CASINO HELP SYSTEM
 ==================

 Keyword : neu
 Title   : Number of up electrons
 Type    : Integer
 Level   : Basic

 DESCRIPTION
 -----------
 For real systems containing atoms, NEU is the total number of spin-up
 electrons referenced by the many-body wave function (for periodic systems,
 this is the number of spin-up electrons in the simulation cell, rather than
 the underlying primitive cell). The number of spin-down electrons is given by
 the keyword NED.

 Note that in the presence of addition or subtraction excitations, NEU refers
 to the state of the system AFTER the required number of electrons have been
 added or removed.  For model electron(-hole) phases such as the HEG, set NEU
 to zero and use the FREE_PARTICLES block to define the number of spin-up
 electrons.
```

# Geometry and Orbital File: “`xwfn.data`”

- `xwfn.data` = `pwfn.data` (plane waves), `bwfn.data` (blips), `gwfn.data` (Gaussians), `stowfn.data` (Slaters) or `awfn.data` (numerical atomic orbitals).

- `xwfn.data` contains the geometry, basis-set, orbital and determinant data produced by the wave-function generating code.

- These files are produced by routines embedded within the external program (e.g., CASTEP, ABINIT and PWSCF) or by various utilities that read the output of the external program (e.g., GAUSSIAN09 and CRYSTAL14) and transform it into a format understandable by CASINO.

- `xwfn.data` is not necessary if the trial wave function can be specified without the use of an external program (e.g., the homogeneous electron gas).
  - In such cases, the system is specified in the `input` file through the **free_particles** block.

# Optimisable Wave-Function File: “correlation.data”

- The correlation.data file contains all the optimisable parameters in the trial wave function.
  - CASINO may improve upon the wave function produced by the generating code by optimising these parameters.

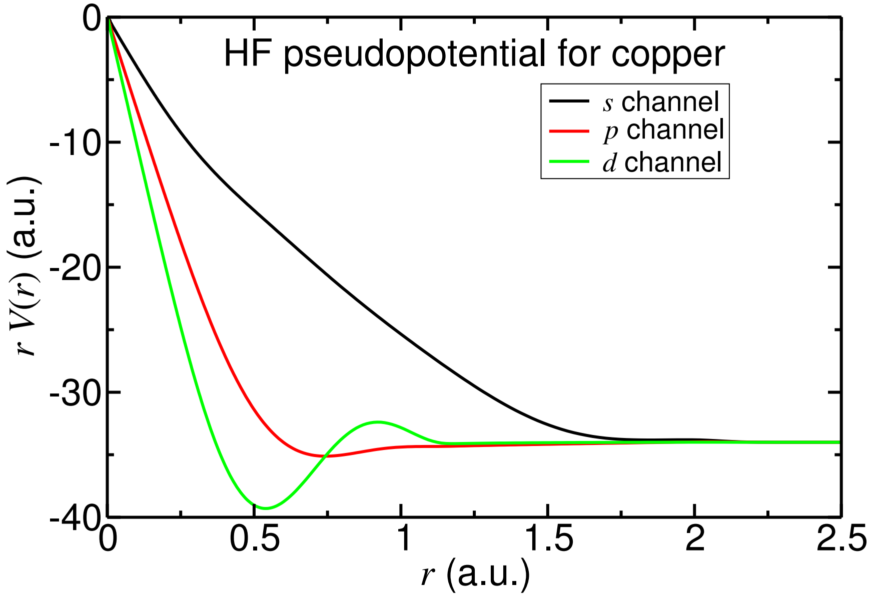
- Specifically, correlation.data may contain any of the following:
  - A Jastrow factor
  - A backflow function
  - Determinant expansion coefficients
  - Parameters modifying the shape of the given DFT/HF orbitals
  - Parameters relating to electron gases and pairing wave functions

- The most common practice is to use a Jastrow factor only.

- Wave-function optimisation leads to the generation of correlation.out.x files. Your chosen wave-function file should then be renamed as correlation.data.

# Pseudopotential File(s): "`x_pp.data`"

- An `x_pp.data` file contains a specification of a nonlocal (angular-momentum-dependent) pseudopotential for element `x`.
  - "`x`" is a chemical symbol in lower case, e.g., "`si`" or "`n`".
  - Contains a radial grid and data for the $s$, $p$ and $d$ angular-momentum channels (higher angular-momentum channels can also be given).



- Each atom can be treated as all-electron or the core can be replaced by a pseudopotential. For all-electron atoms, simply omit the pseudopotential file.

- Get Hartree–Fock pseudopotentials for particular elements from the CASINO pseudopotential library:

```
http://vallico.net/casinoqmc/pplib/
```

# Other Input Files

**config.in & config.out** These store the current state of the calculation (particle positions, energy data, the state of the random number generator, etc.).

- This allows one to continue an interrupted run (see the **newrun** keyword; rename the checkpoint file as `config.in`). By default `config.out` is automatically written after every "block" of moves.
- The `input` keyword **checkpoint** may be used to control the frequency of this (type *casinohelp checkpoint* for information).

**mpc.data** This optional file contains the Fourier components of the charge density corresponding to the Slater wave function and the Fourier components of the $1/r$ Coulomb interaction treated within the minimum-image convention.

- It is only required for the model periodic Coulomb (MPC) interaction used in periodic systems to reduce finite-size effects.

**expot.data** Contains specification of an external potential (e.g., potential for inhomogeneous-electron-gas calculations, or the potential due to an external electric field), together with any data defining the orbitals associated with the potential.

# Main Output File: `out`

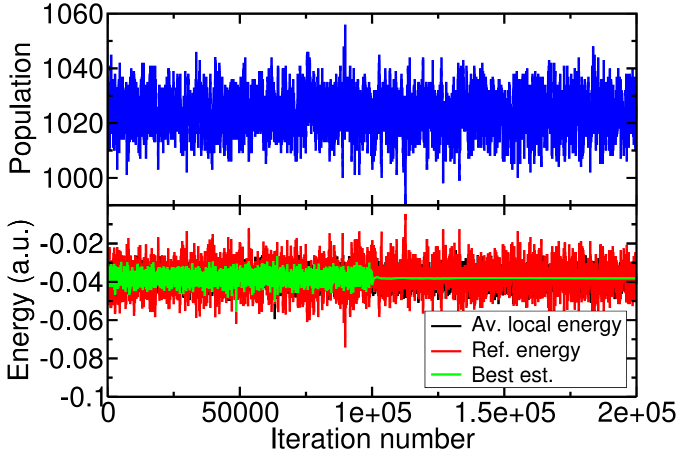
- The `out` file:
  - lists the input parameters (showing defaults),
  - reports the geometry,
  - reports on the setup of the orbitals, Jastrow factor, pseudopotentials, finite-size corrections, etc.
  - reports the progress of VMC / DMC / optimisation,
  - reports simple averages of the local-energy data, etc.
  - reports the results of on-the-fly reblocking of the energy components.

- You should always check the `out` file for **warning** messages.

- The *envmc* script can be used to grab VMC results from the `out` file, including the different energies obtained during wave-function optimisation.

# Other Output Files (I)

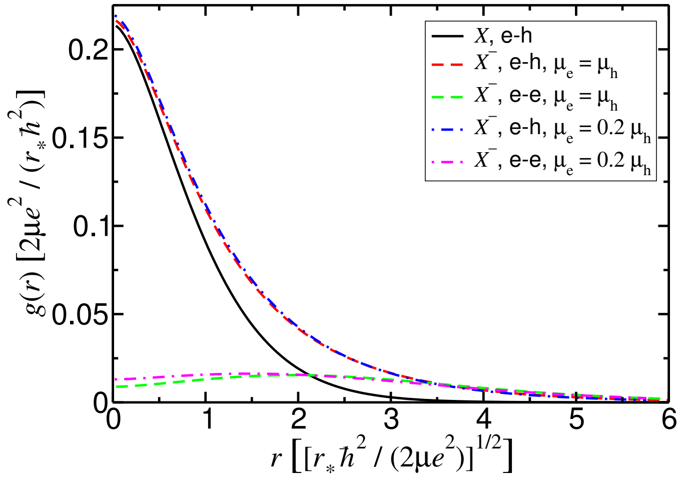
**vmc.hist & dmc.hist** Energy components and important parameters at each iteration of a VMC or DMC simulation (averaged over processes and configurations).



- Components can be plotted using the *plot_hist* utility or summarised using the *graphdmc* utility.
- Statistical analysis of the data in both files can be performed after the calculation using *reblock*. Reblocking is also done “on the fly” by CASINO and the results are written to the end of the out file.

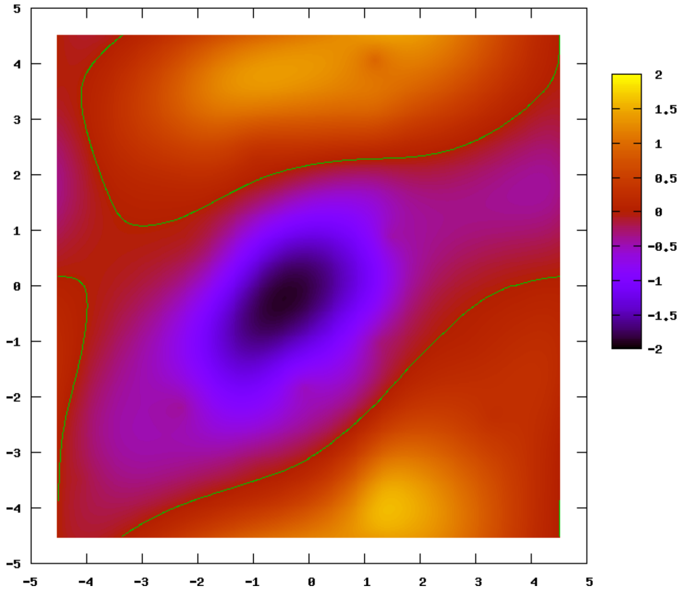
**expval.data** This file may contain data sets describing charge density, spin density, etc.

- If a data set is present and CASINO is asked to accumulate data for that particular set, the newly accumulated data will be added to the existing data.



# Other Output Files (II)

- CASINO can produce plots of the local energy, etc., in `lineplot.dat`, `2Dplot.dat` and `3Dplot.dat` files.
  - 1D plots are best visualised using *Grace*, while 2D/3D plots can be visualised quickly using the CASINO utility *plot_2D*, which provides a user-friendly interface to *gnuplot*.



# Downloading CASINO

- Get the code by filling in the form at:

```
http://vallico.net/casinoqmc/download-casino
```

- Mike Towler will then send you a “CASINO login ID” giving you access to the requested services. This includes the discussion forum, the CASINO website and future downloads from:

```
http://vallico.net/casinoqmc/update-casino
```

- The form also allows you to request “developer status” (given some justification and evidence of some level of competence).

# Unpacking the Distribution

- Change to the directory in which you want the distribution to live (this is usually your home directory).

- Remove/rename any existing `CASINO` directory.

- Unpack the `CASINO_vxxx.tar.gz` distribution (*tar -xzf CASINO_vxxx.tar.gz*).

  – This will result in a new directory called `CASINO` containing the CASINO distribution.

# Installing CASINO

- Operating system requirements:
  - CASINO is designed to run under Linux/Unix (in principle including Macs).
  - CASINO can also be run on Windows machines if the *Cygwin* software is installed, which provides a Linux-like environment. See the manual and FAQ.

- Ensure that your machine has all the relevant software installed: Fortran, C and C++ compilers, an MPI library, (optionally) BLAS/LAPACK, *Grace* and *gnuplot*.
  - For machines administered by other people, this should have been done for you.
  - The install script can help you with installing required software using the package manager (choose the "[i]" option).

- CASINO determines what kind of computer you are running on by looking at the value of the environment variable **CASINO_ARCH**.
  - This tells CASINO to look into a particular "arch" file, which contains instructions that the *make* command and CASINO utilities can follow.
  - You can set this up either manually or automatically, for either currently supported architectures/machines or new, unsupported ones.

# Installation: the *install* Script

- Type *./install* in the base directory of the CASINO distribution, after which you should follow the prompts.
  - You can rerun this installer if you wish to amend your configuration.
- The install script will present you with the following options:

```
[c] Compile CASINO for already-configured CASINO_ARCHs
[s] Sort/remove configured CASINO_ARCHs
[a] Auto-detect valid CASINO_ARCHs for this machine
[p] Pick a specific CASINO_ARCHs for this machine
[n] Create a new CASINO_ARCH for this machine interactively
[y] Install CASINO syntax highlighting for various text editors
[i] Install required software using package manager (requires root access)
[r] Restore the CASINO distribution directory to its original state
[q] Save configuration and quit the installer
[x] Quit the installer without saving
```

# Finding or Creating an “Arch” File

- If you know which **CASINO_ARCH** you want, type its name after selecting the [p] option.

- Otherwise, try “Auto detect” [a]. The *install* script may detect a match for the machine on which you’re running. The script can also suggest generic similar machines on which you can base your personalised installation.

- Choosing [n] will guide you through the procedure for creating your own personalised arch file. Your new arch files can be emailed to the developers for permanent inclusion in the CASINO distribution.

- Multiple alternative configurations are supported. This includes setting up CASINO for use with different compilers on the same machine. You can also create setups in which different machines share the installation directory.

- Once you have all your **CASINO_ARCH**s defined, you can use the [s] option to sort them into a preferred order.

# Compiling CASINO

- Once you have sorted out the arch file, you can compile the code and the utilities by selecting the [c] option. The script will offer a list of defined **CASINO_ARCH**s, e.g.:

```
The following CASINO_ARCHs are configured (it is possible that not all of them
can be compiled from this machine depending on your set-up):
[1] linuxpc-gcc-pbs-parallel.titan
[2] linuxpc-cray-pbs-parallel.titan
[3] linuxpc-ifort-pbs-parallel.titan
At the prompt below enter the numbers corresponding to the CASINO_ARCHs you
would like to compile, separated by spaces.
You can specify which optional compile-time features to enable appending
':<feature>' to each number. Available <features> include:
- 'Openmp' for building OpenMP support
- 'Shm' for building the SMP shared-memory facility (of most use for
calculations with blip or Gaussian basis sets)
- 'OpenmpShm' for building a version with both features enabled
```

- In this case, type *1* to compile with the gcc compiler, *2* to compile with the Cray compiler, etc. Compile special versions, such as the shared-memory version of CASINO, by typing, e.g., *1:Shm*.

# Running CASINO: *runqmc*

- Run CASINO using the *runqmc* script.
  - *runqmc* allows you to run CASINO calculations by entering just one command, which is essentially the same on all kinds of computer.
  - This script uses the arch file to determine how to run jobs on your machine: it even loads modules, writes batch scripts, and submits jobs for you.
  - It detects common errors in input files. Such faults are thus detected immediately rather than when CASINO starts running, which may be many hours or even days later on a machine with a busy queue.
  - To run CASINO in the current directory, simply type *runqmc*. This will occupy all cores on a workstation, or the maximum permitted allocation (in both number of cores and walltime) on a cluster.
  - To specify other calculation parameters, there are various options available. E.g., *runqmc --nproc=224256 --walltime=10h --shm* runs CASINO using 224256 MPI processes with shared memory within each node and a walltime limit of 10 hours. Equivalently, you can type *runqmc -p 224256 -T 10h -s*.
  - For more information, see the manual or type *runqmc --help*.

# Documentation and Sources of Information (I)

- CASINO's **manual**
  - Can be found at `CASINO/manual/casino_manual.pdf` in the distribution.
  - 266 pages of gripping reading about (i) how to use CASINO and (ii) technical details about the implementation of QMC algorithms in CASINO.

- ***casinohelp*** (for `input` keywords)
  - Type, e.g., *casinohelp neu* to find out about the **neu** `input` keyword.

- `README` **and** `FAQ` **files**
  - `README` files are scattered throughout the CASINO distribution.
  - If you are puzzled about a CASINO utility, try looking for the appropriate `README` in the `CASINO/utils` directory; they occasionally contain useful information.
  - "Frequently asked questions" (with answers) can be found at `CASINO/FAQ`.

- **Development log**
  - Found at `CASINO/DIARY` (and online). Record of changes to CASINO since 1999.

# Documentation and Sources of Information (II)

- **Website**

```
http://vallico.net/casinoqmc/
```

  - Includes pseudopotential library, development log, and site from which CASINO can be downloaded.

- **Examples directory**
  - A large number of examples of CASINO input files for different systems can be found in `CASINO/examples`.
  - Note that these calculations are often deliberately cheap so that the wave-function files are not too large and the examples do not take hours or days to run.
  - *So please be aware that real calculations may require larger plane-wave cutoff energies, larger numbers of steps, larger numbers of configurations, etc.*

- **Documents directory**
  - `CASINO/doc` contains a random assortment of documents related to CASINO.

# Documentation and Sources of Information (III)

- Many features of CASINO are described in **published articles**, which are often the most detailed and accurate description of the features.:
  - CASINO's Jastrow factor: N.D. Drummond *et al.*, Phys. Rev. B **70**, 235119 (2004).
  - General Jastrow factor: P. López Ríos *et al.*, Phys. Rev. E **86**, 036703 (2012).
  - Backflow (CASINO): P. López Ríos *et al.*, Phys. Rev. E **74**, 066701 (2006).
  - Variance minimisation: C.J. Umrigar *et al.*, Phys. Rev. Lett. **60**, 1719 (1988).
  - Unreweighted variance minimisation: N.D. Drummond and R.J. Needs, Phys. Rev. B **72**, 085124 (2005).
  - Energy minimisation: C.J. Umrigar *et al.*, Phys. Rev. Lett. **98**, 110201 (2007).
  - DMC: D.M. Ceperley and B.J. Alder, Phys. Rev. Lett. **45**, 566 (1980); C.J. Umrigar *et al.*, J. Chem. Phys. **99**, 2865 (1993).
  - Blip basis functions: D. Alfè and M.J. Gillan, Phys. Rev. B **70**, 161101 (2004).
  - Gaussian cusp corrections: A. Ma *et al.*, J. Chem. Phys. **122**, 224322 (2005)
  - Finite-size corrections: S. Chiesa *et al.*, Phys. Rev. Lett. **97**, 076404 (2006); N.D. Drummond *et al.*, Phys. Rev. B **78**, 125106 (2008).
  - CASINO pseudopotential library: J.R. Trail and R.J. Needs, J. Chem. Phys. **122**, 014112 (2005); J. Chem. Phys. **122**, 174109 (2005).

# Sources of Support

- **Discussion forum** (first instance):

  ```
  http://www.vallico.net/casino-forum/
  ```

  - To make this work, please supply answers to questions where you can!

- **Developers** (for more technical questions or bug reports):
  - Mike Towler (`mdt26` at `cam.ac.uk`).
  - Neil Drummond (`n.drummond` at `lancaster.ac.uk`).
  - Pablo López Ríos (`pl275` at `cam.ac.uk`).