

## Version 1.0 de la bibliothèque Q5Cost

*Anthony Scemama*, Antonio Monari, Celestino Angeli, Stefano Borini, Stefano Evangelisti, and Elda Rossi

LCPQ-IRSAMC CNRS, Toulouse (France)  
scemama@irsamc.ups-tlse.fr

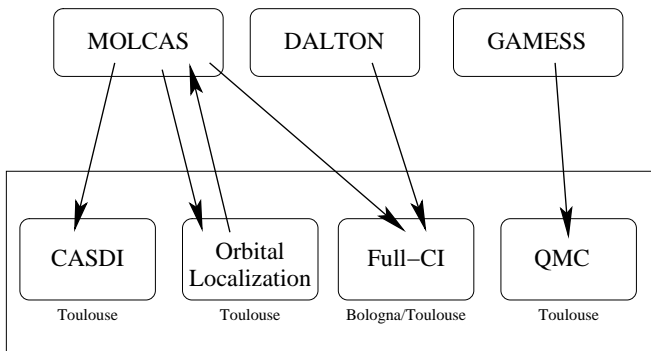
20 Novembre 2008

# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

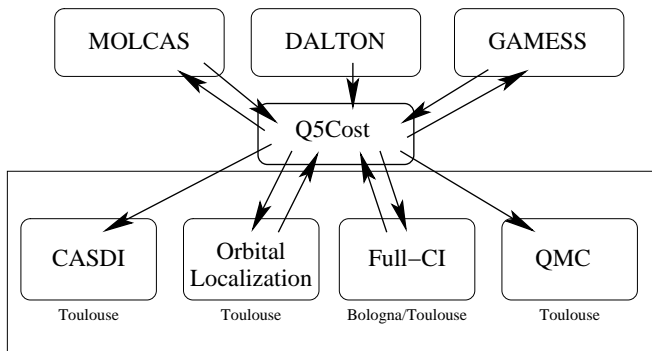
# Le besoin d'un format de fichier commun

Situation à Toulouse il y a quelques années...



# Le besoin d'un format de fichier commun

Situation à Toulouse aujourd'hui



# Présentation de l'exposé

- 1 **Contexte**
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

# Motivations

- Nos codes lisent des fichiers produits par d'autres codes, dont le format peut changer d'une version à l'autre.
- La structure de ces fichiers est parfois difficile à comprendre
- Il faut avoir autant d'interfaces que de paires de codes ( $N(N - 1)$ )
- Avec un format de fichier commun, il faut autant d'interfaces que de codes ( $N$ )

# Motivations

- Nos codes lisent des fichiers produits par d'autres codes, dont le format peut changer d'une version à l'autre.
- La structure de ces fichiers est parfois difficile à comprendre
- Il faut avoir autant d'interfaces que de paires de codes ( $N(N - 1)$ )
- Avec un format de fichier commun, il faut autant d'interfaces que de codes ( $N$ )

# Motivations

- Nos codes lisent des fichiers produits par d'autres codes, dont le format peut changer d'une version à l'autre.
- La structure de ces fichiers est parfois difficile à comprendre
- Il faut avoir autant d'interfaces que de paires de codes ( $N(N - 1)$ )
- Avec un format de fichier commun, il faut autant d'interfaces que de codes ( $N$ )



# Motivations

- Nos codes lisent des fichiers produits par d'autres codes, dont le format peut changer d'une version à l'autre.
- La structure de ces fichiers est parfois difficile à comprendre
- Il faut avoir autant d'interfaces que de paires de codes ( $N(N - 1)$ )
- Avec un format de fichier commun, il faut autant d'interfaces que de codes ( $N$ )

# But

Pour les codes de chimie quantique, on a besoin de :

- Fichiers structurés (facilement compréhensibles)
- Fichiers indépendants de l'architecture
- Fichiers compressés
- Un accès facile aux données en Fortran

# But

Pour les codes de chimie quantique, on a besoin de :

- Fichiers structurés (facilement compréhensibles)
- Fichiers indépendants de l'architecture
- Fichiers compressés
- Un accès facile aux données en Fortran

# But

Pour les codes de chimie quantique, on a besoin de :

- Fichiers structurés (facilement compréhensibles)
- Fichiers indépendants de l'architecture
- Fichiers compressés
- Un accès facile aux données en Fortran

# But

Pour les codes de chimie quantique, on a besoin de :

- Fichiers structurés (facilement compréhensibles)
- Fichiers indépendants de l'architecture
- Fichiers compressés
- Un accès facile aux données en Fortran

# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

# Qu'est-ce que Q5Cost ?

- Un format de fichier adapté à la chimie quantique
- Une API en Fortran pour accéder aux données
- Quelques outils pour manipuler ces fichiers

# Qu'est-ce que Q5Cost ?

- Un format de fichier adapté à la chimie quantique
- Une API en Fortran pour accéder aux données
- Quelques outils pour manipuler ces fichiers



# Qu'est-ce que Q5Cost ?

- Un format de fichier adapté à la chimie quantique
- Une API en Fortran pour accéder aux données
- Quelques outils pour manipuler ces fichiers

# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 **Q5Cost**
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

# Le format de fichier

## Q5Cost utilise la technologie HDF5

- Fichiers binaires
- Structurés comme dans un système de fichiers
- Indépendants de l'architecture
- Compression (avec zlib)
- Excellentes performances pour les I/O

# Le format de fichier

Q5Cost utilise la technologie HDF5

- Fichiers binaires
  - Structurés comme dans un système de fichiers
  - Indépendants de l'architecture
  - Compression (avec zlib)
  - Excellentes performances pour les I/O

# Le format de fichier

Q5Cost utilise la technologie HDF5

- Fichiers binaires
- Structurés comme dans un système de fichiers
- Indépendants de l'architecture
- Compression (avec zlib)
- Excellentes performances pour les I/O

# Le format de fichier

Q5Cost utilise la technologie HDF5

- Fichiers binaires
- Structurés comme dans un système de fichiers
- Indépendants de l'architecture
- Compression (avec zlib)
- Excellentes performances pour les I/O

# Le format de fichier

Q5Cost utilise la technologie HDF5

- Fichiers binaires
- Structurés comme dans un système de fichiers
- Indépendants de l'architecture
- Compression (avec zlib)
- Excellentes performances pour les I/O

# Le format de fichier

Q5Cost utilise la technologie HDF5

- Fichiers binaires
- Structurés comme dans un système de fichiers
- Indépendants de l'architecture
- Compression (avec zlib)
- Excellentes performances pour les I/O



# Le format de fichier

## 5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients, ...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Le format de fichier

5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients, ...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Le format de fichier

5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients, ...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Le format de fichier

5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients, ...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Le format de fichier

5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients,...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Le format de fichier

5 groupes principaux :

- System (le système moléculaire)
  - Geometrie moléculaire, symétrie
  - Énergie de répulsion nucléaire, nb d'électrons  $\alpha$  et  $\beta$
- Basis (la base atomique)
  - Système de coordonnées (sphérique/cartésien)
  - Contractions de gaussiennes (exposants, coefficients,...)
- AO (Infos sur les orbitales atomiques)
  - LCAO de symétrie adaptée
  - Intégrales mono- et bi-électroniques, matrice de recouvrement
- MO (Orbitales moléculaires)
  - Valeurs propres, nb d'occupations, symétrie, ...
  - Classification (frozen, active, virtual, alpha, beta)
  - Coefficients
- WF (Fonction d'onde)
  - Déterminants et coefficients
  - Énergie totale

# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5

- Tous les noms de routine peuvent être calculés, par exemple :

```
Q5Cost_System_get_num_alpha(file_id,num_alpha,error)  
--1--- --2--- -3- ----4---- ---5--- ----6---- --7--
```

- 1) Tous les noms de routines commencent par "Q5Cost"
- 2) Le groupe qui contient les données
- 3) set/get
- 4) Le nom de la donnée à atteindre
- 5) L'identificateur du fichier
- 6) La variable dans laquelle se trouve(ra) la donnée
- 7) Un code d'erreur. 0 si succès



# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès

# L'API Q5Cost

- Un ensemble de routines Fortran qui encapsulent les appels HDF5 → Les utilisateurs n'ont pas besoin de connaître HDF5
- Tous les noms de routine peuvent être calculés, par exemple :  
`Q5Cost_System_get_num_alpha(file_id,num_alpha,error)`  
--1--- --2--- -3- ----4----- --5--- ----6----- --7--
  - 1) Tous les noms de routines commencent par "Q5Cost"
  - 2) Le groupe qui contient les données
  - 3) set/get
  - 4) Le nom de la donnée à atteindre
  - 5) L'identificateur du fichier
  - 6) La variable dans laquelle se trouve(ra) la donnée
  - 7) Un code d'erreur. 0 si succès



# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

# The package

- `./configure`
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- `./configure`
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- ./configure
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- ./configure
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- ./configure
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- ./configure
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump

# The package

- ./configure
- bibliothèque et fichiers d'include (\*.mod)
- tests
- documentation (format de fichier et API)
- Liens avec F77, C++ et Python générés automatiquement
- q5edit (interactif)
- q5dump



# Présentation de l'exposé

- 1 Contexte
  - Motivations
- 2 Q5Cost
  - Le format de fichier
  - L'API Q5Cost
  - Le package Q5Cost
- 3 Conclusion

## Quelques codes qui communiquent actuellement

- GAMESS
- Dalton
- Molcas
- Full-CI
- DOLO
- QMC=Chem
- Columbus (Vienne)
- Aces II (Budapest)
- Molekel

## Remerciements

E. Rossi, A. Emerson, G. Bendazzoli, A. Monari, S. Evangelisti, D. Maynau, M. Kállay, K. Baldrige, K. Ruud, R. Cimiraglia, C. Angeli, C. Pastore, V. Vallet , J.-P. Flament, A. Scemama, J. Sanchez-Marin , P. Szalay, A. Tajti, S. Borini, and more...  
Ce projet est partiellement financé par le projet européen COST-D37.