5 State of the Art tools

In the Section 2 we illustrated the background knowledge needed to understand the issues presented in the Section 3. The subject of this section is how stakeholders tackle these issues. First we show a little more in detail some examples of real QSAR models, giving details on how these models deal in practice with the complexity of toxic mechanisms. Then we make an overview on the available software, and in particular we focus on CDK Java Libraries, Open Source Java librarie for Computational Chmistry that will be extremely useful in our solution. The section is concluded by a list of websites where it is possible for firms to collect information on REACH regulation.

5.1 QSAR models in practice

In this section we give some examples of QSAR models developed in the last 10 years. These examples will give a more detailed idea of how these model actually works, and of the range of possible endpoints.

The first model we present is a model for quail dietary toxicity, presented in [30]. The considered dataset refers to quail dietary exposure to different kinds of pesticides (chlorinated compounds, thiazines, organophosphates..), whose toxicity is measured as log(1/LC), where LC stands for LC50-96h, that is the concentration that kills half of the quail population in 96 hours. In other words, the examined endpoint is death.

Atomic orbital graphs are used to describe the chemical structure of the pesticides, instead of standard molecular graphs, to take into account atomic orbitals, such as $1s^1$,$2p^2$, $3d^{10}$; this further information is mathematically expressed through a descriptor called ${}^0X_{cw}$, a particular invariant for these graphs. A linear model is obtained through Monte Carlo optimization of the value of the descriptor and least squares method, leading to an expression of toxicity as:

$$log(1/LC) = C_0 + C_1 \cdot X_{CW}$$

Validation on a test set shows a reasonable agreement with experimental data ($R^2 \approx 0.65$), considered that a single model here is forced to predict activity of pesticides with a number of different (and unknown) toxic mechanisms: models like this one are called global models.

Other global models can be found in [34]. In this paper the endpoint is bioconcentration, that is the process of accumulation of chemicals by aquatic organisms (e.g. fishes) through non-dietary routes, such as cutaneous absorption. Bioconcentration is measured as a bioconcentration factor (BCF), that is the ratio between the concentration of a given chemical in the organism and the steady concentration of the same chemical in the environment. Authors use topological, constitutional and functional groups descriptors to develop a number of global models: linear regression models as the one in the previous example, and non linear models through radial basis function neural networks. We recall that for radial basis function neural networks the final model is:

$$log(BCF) = \sum_j w_i h_j(x) + b,$$

where j are the neurons in the hidden layer, whose activation function is:

$$h_j(x) = exp(-\|x - c_j\|^2 / r_j^2),$$

that is a Gaussian function centred in c_j with width r_j. All these models show a reasonable predictive power, as R^2 stays between 0.74 and 0.80 for all models.

Anyhow, global models in general cannot achieve really high predictive results, as it is unlikely that such complex phenomena as toxic mechanisms are well described by a unique function of some descriptors, regardless of its complexity. To further improve predictive power, one has to build multi step models, that start with local descriptions of the data space and then put together these local information in a suitable way. This is the idea of the so called hybrid models, or expert systems.

A first example of hybrid models is found once again in [34]. In this approach to expert systems, the underlying hypothesis is that every single global model catches different pieces of information stored in the data; therefore a winning strategy to estimate the endpoint could be to run all the different models, and then to perform an “ensambling”/ “averaging” of the different results.

In the article the authors follow these steps:

1. run the global models;
2. average the results;
3. divide the range of outcomes in some areas, say three areas;
4. build in each area the final predictive model, that will be:
   $$log(BCF) = C_0 + C_1 \cdot operator(y_1 ,.., y_k)$$
   where y_k is the endpoint estimate by the k-th model, and operator can be min, max, mean.

Another possible strategy is to build expert systems that first divide the information space according to the biological rationale of the studied phenomenon, then build a proper model for each sub-domain and finally link every chemical to the most suitable sub-domain, and hence to the best model.

An example of this procedure can be found in [26], in QSAR applied to acute toxicity for fishes. First, the authors individuate eight different modes of action: base line narcosis or narcosis I, polar narcosis or narcosis II, ester narcosis or narcosis III, oxidative phosphorylation uncoupling, respiratory inhibition, electrophile/proelectrophile reactivity, AChE inhibition and CNS seizure response.

As a second step, every chemical of the considered data set (617 chemicals) is assigned to a single mode of action, through both an experimental procedure and a review of the existing literature. We remember that this implies killing a statistically significant number of fishes for each chemical, measuring until death occurs a number of variables such as heart rate, blood pH, hematocrit and symptoms such as convulsions, spasms, reaction to stimuli, body coloration.

After this splitting procedure, a QSAR linear regression is performed for each class and finally an heuristic expert system that assigns every new chemical considered to its (predicted) mode of action is built. This expert system looks for those molecular fragments that are specific for each mode of action; thus it is implemented as a set of conditional statements that will be used to classify chemicals: for example *"if a chemical has an oxygen attached to an aromatic ring and the number of halogens is less than or equal to 2, then the associated mode of action is narcosis II"*. Specific fragments for each mode of action were identified "a priori" by human knowledge, but they could have been computationally assessed as well. Validation on a test set gives $R^2 = 0.91$, that is to say an excellent predictive power.

It is important to underline that each mode of action is not related to a single toxic molecular mechanism. For example, CNS seizure agents and respiratory inhibitors can act through a variety of receptors (see [8] and [1]). Moreover, the experimental assessment procedure shows that standard chemical classification is not suited for modes of action. For example, ethers are usually associated to narcosis I mode of action, but the experimental assessment shows that ethers may also act as electrophiles; conversely, compounds from classes that are not supposed to be narcosis I type act in this way.

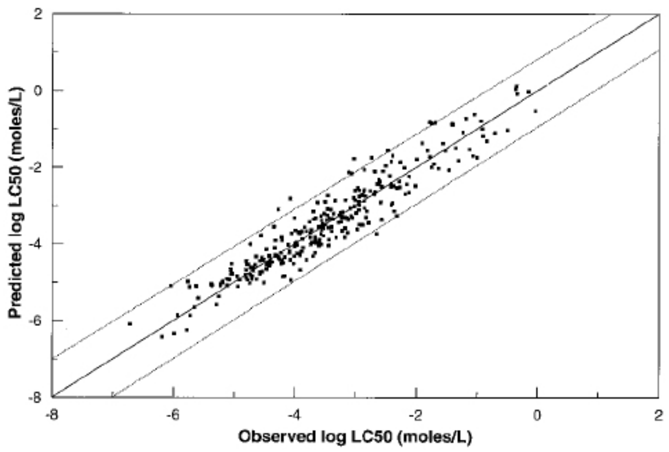

Figure 19: predicted LC50 vs. observed LC50 in [26]. The ideal plot would be with all points on the unity line (central line). The upper and lower bound represent plus or minus one log unit from unity.

Classification based on mode of action analyses therefore seems to be the most appropriated choice, but even this approach is arguable. The first issue is that modes of action are not uniquely defined. The second issue is that modes of action should be considered as continuum, as they are the visible outcome of several cooperating or competitive molecular mechanisms rather than linked to a single one, as we already pointed out.

As a consequence, also chemical classification based clustering may be successfully used in expert system QSAR building. An example is [18]. In this paper, that refers to the same data set of [26], local linear models are built on 13 subsets obtained splitting chemicals according to the E.P.A. classification: hydrocarbons, ethers, alcohols, aldehydes, ketones, acids, nitriles and sulphur compounds, amines, benzenes, phenols, heterocyclics, carbmates and other pesticides, and the remaining chemicals were merged in a residual class.

In this case, the classifier does not choose the most likely mode of action, but tries to identify the most fruitful model, on the basis of an automatic selection of nominal chemical classes. The appropriated class is not selected through standard definition ("human knowledge") nor heuristics conditional statement, but once again using constitutional descriptors.

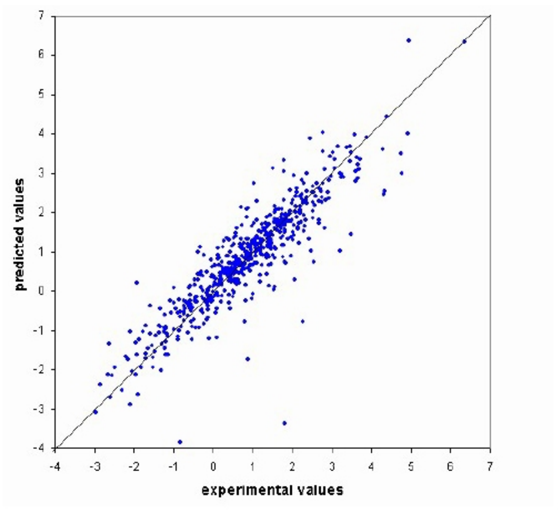

Figure 21: Predicted values vs. experimental values from expert system in [18]

In both [26] and [18], the expert systems show a remarkable improvement of predicting power with respect to the single global models. In addition to this noticeable feature, expert systems are attractive also because they lead to a reduction of the needed computational time, as the time consuming algorithms that tune the parameters of the models are applied to small set of data (the sub-domains) rather than to the whole dataset.

5.2 Available software

5.2.1 Computational tools for applying QSARs

A wide variety of publicly available and commercial computational tools have been developed that are suitable for the development and application of QSARs. Such tools include methods for range of QSAR-related tasks, including data management and data mining, descriptor generation, molecular similarity analysis, analogue searching and hazard assessment.

Among these tools, QSAR-based expert systems enable predictions of chemical toxicity to be obtained directly from chemical structure. All are built upon some experimental toxicity data with rules derived from the data[2,3].

A list of the most used tools/databases publicly available will be shown below. In annex I, a list of reference website is also provided.

Ambit is freely available software for data management and QSAR applications, including searchable databases and tools for grouping and applicability domain assessment. The *AMBIT* database stores chemical structures, their identifiers such as CAS, INChI numbers, attributes such as molecular descriptors, experimental data together with test descriptions, and literature references. The database can also store QSAR models. In addition, the software can generate a suite of 2D and 3D molecular descriptors.

Toxtree, developed by Ideaconsult Ltd under contract to ECB, is a freely available application which is able to estimate different types of toxic hazard by applying structural rules. Currently, plug-ins are available for applying the following rulebases: a) the Cramer classification scheme for TTC (Threshold of Toxicological Concern) estimation; b) the Verhaar scheme for predicting the mode of toxic action in aquatic species; c) decision trees for estimating skin and eye irritation and corrosion potential, based on the BfR rules, and d) the Benigni-Bossa rulebase for mutagenicity and carcinogenicity.

The *JRC QSAR Model Database*, which is currently under development will be a searchable tool for linking chemicals of interest to a collection of robust summaries of (Q)SAR models. The summaries are being compiled by using a standard (Q)SAR Model Reporting Format (QMRF). A database with a web-based interface will be implemented to allow on-line access to the JRC QSAR Model Database.

The *EPI (Estimation Program Interface) Suite* program integrates a number of estimation models for the prediction of environmental and physical/chemical properties in one convenient interface.

These models include KowWin (for estimating log Kow), AopWin (for predicting gas-phase reaction rates), HenryWin (for Henry's Law constant), MPBPVP (for predicting melting point, boiling point, and vapour pressure), WsKow (for estimating water solubility and log Kow), Hydro (for estimating hydrolysis rate constants for specific organic classes), DermWin (for estimating the dermal permeability coefficient (Kp)), ECOSAR (described above) and BCFWin (for estimating the bioconcentration factor). *EPI Suite* is freely available from the US-EPA website.

Referring to commercially available tools other two examples can be mentioned.

TOPKAT is a statistical system developed by Accelrys, Inc consisting of a suite of QSAR models for a range of different endpoints. There are currently 16 modules for the following endpoints: aerobic biodegradability, Ames mutagenicity, Daphnia magna EC50, developmental toxicity, fathead minnow LC50, FDA rodent carcinogenicity, NTP rodent carcinogenicity ocular irritancy, logKow, rabbit skin irritancy, rat chronic LOAEL, rat inhalation toxicity LC50, rat Maximum Tolerated Dose (MTD), rat oral LD50, skin sensitisation, and Weight of Evidence rodent carcinogenicity.

TOPKAT models are typically based on the analysis of large datasets of toxicological information derived from the literature. The molecular descriptors used include structural (e.g. molecular bulk, shape, symmetry), topological and electrotopological indices. The QSARs are developed by regression analysis for continuous endpoints and by discriminant analysis for categorical data. It estimates the confidence in the prediction by applying the patented Optimal Predictive Space (OPS) validation method.

TerraQSAR™ is a collection of computation programs for the prediction of biological effects and physico-chemical properties of organic compounds. The available models developed using a probabilistic neural network (PNN) methodology include: DM 24 hr EC50 for Daphnia magna, E2-RBA estrogen receptor binding affinity (RBA), FHM 96-h LC50 for fathead minnow (Pimephales promelas), log P octanol/water partition coefficient, etc.

A number of commercial software programmes have been developed for the calculation of the molecular descriptor. Some of them are provided below. The list of reference website is in Annex I.

Accord for Excel uses Accord Chemistry Engine to handle chemical structures and incorporates a number of add-ins to perform chemical calculations based on the structures of a compound in a record.

ADAPT is a QSAR toolkit with descriptor generation (topological, geometrical, electronic and physico-chemical descriptors), variable selection, regression and artificial neural network modelling.

CODESSA has been developed for the calculation of several topological, geometrical, constitutional, thermodynamic, electrostatic and quantum-chemical descriptors. It also includes tools for regression modelling and variable selection.

DRAGON has been developed for the calculation of several sets of molecular descriptors from molecular geometries (topological, geometrical, WHIM, 3D-MoRSE, molecular profiles).

GRIN/GRID calculates the GRID empirical force field at grid point.

HYBOT-PLUS has been developed for the calculation of hydrogen bond and free energy factors.

MOLCONN-Z, successor to *MOLCONN-X*, *MOLCONN-Z*, calculates the most known topological descriptors, including electrotopological and orthogonalised indices.

Last release: 3.0.

OASIS has been developed for the calculation of steric, electronic and hydrophobic descriptors.

POLLY has been developed for the calculation of topological connectivity indices.

SYBYL/QSAR has been developed for the calculation of EVA descriptors, CoMFA and CoMSIA fields. It also includes several QSAR tools.

TSAR is characterized by statistical and database functions with molecular and substituent property calculations.

5.2.2 Chemistry Development Kit (CDK)

The Chemistry Development Kit (CDK) is a Java library for structural chemo- and bioinformatics, used in over 10 different academic and industrial projects worldwide. As a successor of Christoph Steinbeck's CompChem libraries, the CDK library evolved into a full chemo-informatics package with code reaching from QSAR descriptor calculations to 2D and 3D model building. It is maintained as a SourceForge project under http://www.sourceforge.net/projects/cdk. SourceForge offers bug tracking, mailing lists, support manager and CVS access[3].

The CDK library contains a huge number of various classes and it is impossible to describe all of them. Instead, some of the basic classes needed for most of the calculations, including the calculations of molecular descriptors, will be explained in this section.

The class Atom represents the idea of a chemical atom. The following constructors are available for creating an Atom object:

- **public Atom() –** constructs a completely unset Atom.

[3] See: http://apps.sourceforge.net/mediawiki/cdk

- **public Atom(String elementSymbol)** - Constructs an Atom from a String containing an element symbol.
- **public Atom(String elementSymbol, javax.vecmath.Point3d point3d)** - Constructs an Atom from a String containing an element symbol and an additional Point3d object representing the 3D coordinates of the atom.
- **public Atom(String elementSymbol, javax.vecmath.Point2d point2d)** - Constructs an Atom from a String containing an element symbol and an additional Point2d object representing the 2D coordinates of the atom.

There are also setter and getter methods for the partial charge, the hydrogen count, the stereo parity and the location in a 2D and 3D space of the Atom object.

The class Bond implements the concept of a covalent bond between two atoms. A bond is considered to be the number of electrons connecting two atoms. The following constructors are available for creating a Bond object:

- **public Bond()** - Constructs an empty bond.
- **public Bond(IAtom atom1, IAtom atom2)** - Constructs a bond with a single bond order between the two atoms given as input parameters.
- **public Bond(IAtom atom1, IAtom atom2, double order)** - Constructs a bond with a given bond order between the two atoms given as input parameters.
- **public Bond(IAtom atom1, IAtom atom2, double order, int stereo)** - Constructs a bond with a given order and stereo orientation from an array of atoms.

Some of the more important methods of this class are the following:

- **public void setAtoms(IAtom[] atoms)** - Sets the array of atoms making up this bond.
- **public int getAtomCount()** - Returns the number of Atoms in this Bond.
- **public IAtom getAtom(int position)** - Returns the Atom from this bond at the position given as an input parameter.
- **public IAtom getConnectedAtom(IAtom atom)** - Returns the atom connected to the atom given as an input parameter.
- **public void setAtom(IAtom atom, int position)** - Sets an Atom in this bond at the position given as an input parameter.

There are also setter and getter methods for the order of the bond, the stereo descriptor, the geometric 2D center, and the geometric 3D center.

The class Molecule represents the concept of a chemical molecule, an object composed of atoms connected by bonds. The following constructors are available for creating a Molecule object:

- **public Molecule()** - Creates a Molecule object without Atoms and Bonds.
- **public Molecule(int atomCount, int bondCount, int lonePairCount, int singleElectronCount)** - Constructor a Molecule object where the parameters define the initial capacity of the arrays.
- **public Molecule(IAtomContainer container)** - Constructs a Molecule with a shallow copy of the atoms and bonds of an AtomContainer.

The class AtomContainer is a base class for all chemical objects that maintain a list of Atoms and ElectronContainers. The following constructors are available for creating an AtomContainer object:

- **public AtomContainer()** - Constructs an empty AtomContainer.
- **public AtomContainer(IAtomContainer container)** - Constructs an AtomContainer with a copy of the atoms and electronContainers of another AtomContainer.
- **public AtomContainer(int atomCount, int bondCount, int lpCount, int seCount)** - Constructs an empty AtomContainer that will contain a certain number of atoms, bonds, lone pairs and single electrons. It will set the starting array lengths to the defined values, but will not create any of these objects.

Some of the more important methods of this class are the following:

- **public void setAtoms(IAtom[] atoms)** - Sets the array of atoms of this AtomContainer.
- **public void setAtom(int number, IAtom atom)** - Set the atom at the position given as an input parameter.
- **public IAtom getAtom(int number)** - Gets the atom at the position given as an input parameter.
- **public java.util.Iterator atoms()** - Returns an Iterator for looping over all atoms in this container.
- **public int getAtomNumber(IAtom atom)** - Returns the position of a given atom in the atoms array. It returns -1 if the atom does not exist.
- **public int getAtomCount()** - Returns the number of Atoms in this Container.
- **public List getConnectedAtomsList(IAtom atom)** - Returns an ArrayList of all atoms connected to the given atom.
- **public int getConnectedAtomsCount(IAtom atom)** - Returns the number of atoms connected to the given atom.
- **public void addAtom(IAtom atom)** - Adds an atom to this container.
- **public void removeAtom(int position)** - Removes the atom at the given position from the AtomContainer.
- **public void removeAtom(IAtom atom)** - Removes the given atom from the AtomContainer.
- **public void removeAllElements()** - Removes all atoms and bond from this container.

Analogue methods exist for manipulating the Bond, LonePair and SingleElectron objects.

The class AtomContainerManipulator is a class with convenient methods for manipulating AtomContainer objects. Some of the more important methods of this class which are useful for implementing the descriptor classes involve hydrogen manipulation and are listed below:

- **public static int getTotalHydrogenCount(IAtomContainer atomContainer) –** Returns the summed implicit hydrogens of all atoms in this AtomContainer.
- **public static int countExplicitHydrogens(IAtomContainer atomContainer, IAtom atom) –** Returns the number of explicit hydrogens on the given IAtom.
- **public static int countHydrogens(IAtomContainer atomContainer, IAtom atom) –** Returns the summed implicit and explicit hydrogens of the given IAtom.

- **public static IAtomContainer removeHydrogens(IAtomContainer atomContainer) -** Produces an AtomContainer without explicit hydrogens but with hydrogen count from one with hydrogens. The new molecule without hydrogens is a deep copy.

The SmilesParser class parses a SMILES string and an AtomContainer. It does not parse stereochemical information, but the following features are supported: reaction smiles, partitioned structures, charged atoms, implicit hydrogen count and isotope information. It contains a number of methods but the method of our interest is the one for parsing a SMILES string:

- **public IMolecule parseSmiles(String smiles)** - Parses a SMILES string and returns a Molecule object representing the constitution given in the SMILES string.

In order to get a better understanding of the above mentioned classes, the following simple code segments illustrate their possible use. The first one creates an AtomContainer object and an Atom object from a String containing the element symbol ‘C’. It sets the hydrogen count of this atom to 4 and it adds it to the AtomContainer. The second one creates an AtomContainer object by parsing the SMILES string `"NC(CO)C(=O)O"`.

```
IAtomContainer methan= new AtomContainer();
Atom c=new Atom("C");
c.setHydrogenCount(4);
methan.addAtom(c);

SmilesParser parser=new SmilesParser();
IAtomContainer molecule=parser.parseSmiles("NC(CO)C(=O)O");
```

With the addition of the cdk.qsar module, CDK has been extended to allow for the calculation of molecular descriptors. Currently 33 descriptors are present covering topological, geometric and electronic descriptor classes. The rest of this section is dedicated to the use of CDK for calculating molecular descriptors.

In order to get a better insight into this matter, it is important to understand the structure of the descriptor classes and the way of implementing individual descriptors. All descriptor classes implement the IMolecularDescriptor interface and as such must implement all its methods, namely:

- **DescriptorSpecification getSpecification()** – returns an object containing the descriptor specification.
- **void setParameters(Object params[])** – sets the parameters attribute of the Descriptor object.
- **Object[] getParameters() –** gets the parameters attribute of the Descriptor object.
- **DescriptorValue calculate(IAtomContainer atomContainer) –** calculates the value of the descriptor for the atom/molecule given as an input parameter and returns a DescriptorValue object which contains this value.
- **IDescriptorResult getDescriptorResultType()** – returns an object that implements the IDescriptorResult interface indicating the actual type of values returned by the descriptor in the DescriptorValue object. Depending on the type of the descriptor, the IDescriptorResult can be of one of the following types: BooleanResult, DoubleResult, DoubleArrayResult, IntegerResult or IntegerArrayResult.
- **String[] getParameterNames()** – gets the parameterNames attribute of the Descriptor object.
- **Object getParameterType(String name)** – gets the parameterType attribute of the Descriptor object.

5.3 Overview of interesting websites on REACH

A lot of different websites deal with the main aspects of REACH legislation. They can be divided into two main categories, such as the websites that are completely dedicated to the new regulation system, on the one hand, and the websites which deal with REACH only within one or few sections.

5.3.1 Websites totally dedicated to REACH

There are a lot of websites that are totally dedicated to REACH and some of them will briefly described below. Among these websites there are ones where general information are provided and some others that are made by consulting agencies.

General information websites

http://eur-lex.europa.eu/JOHtml.do?uri=OJ:L:2007:136:SOM:EN:HTML

The official Reach Regulation published on the Official Journal of the European Union, available from the link above.

http://echa.europa.eu/

The ECHA website provides access to technical guidance, frequently asked questions (FAQs), software tools and helpdesks on REACH. Here the latest updates on guidance, tools, data on chemicals and the Regulation can be found.

http://echa.europa.eu/reach/helpdesk/nationalhelp_contact_en.asp

Every European Country has its own helpdesk. The link mentioned refers to web page in ECHA website, where the list of national REACH helpdesks is provided with their specific links.

http://reach.mi.camcom.it

Summaries about REACH in general are available: people involved (“chi”), field of action (“dove”), actions (“come”), time scheduling (“quando”), aims and basic principles (“perchè”).

The website is in Italian.

Consulting services’ websites

http://www.reach-cdrom.eu/

Detailed information and several documents about REACH in general are available on the website.

http://www.denehurst.co.uk/index.html

Detailed information and several documents about REACH in general are available on the website. Specialised consultancy services are provided.

http://reach.itertech.it/index.php

Detailed information and several documents about REACH in general are available on the website. Some consultancy services are provided too.The website is in Italian.

http://www.regolamentoreach.it

Detailed information and several documents about REACH in general are available on the website. Some consultancy services are provided too.

The website is in Italian.

http://www.reachcolours.it

A lot of different solutions and services are provided in order to help firms to pre-register and register the different substances (CAS, EINECS, ELINCS research, classification and labelling of substances, management of a SIEF etc.).

5.3.2 Websites partly dedicated to REACH

Some interesting websites have only one or some sections related to the new regulation for chemical substances and brief list is provided below.

http://ecb.jrc.ec.europa.eu/

http://ecb.jrc.ec.europa.eu/reach/

http://ecb.jrc.ec.europa.eu/qsar/

This website is made by The Consumer Products Safety & Quality (CPS&Q) Unit, formerly known as European Chemicals Bureau (ECB). Here several documents and guidance about REACH can be found. In addition, there is a specific section which is dedicated to QSAR and where some software tools are freely accessible, such as JRC QSAR Model Database, Toxtree, DART, etc.

General information and a pdf file that describes the current version of the QSAR Prediction Reporting Format (QPRF) are also provided.

http://ec.europa.eu/environment/chemicals/reach/reach_intro.htm

The European Commission's Environment Directorate-General (DG) has developed within an interim strategy a number of REACH Implementation Projects (RIPs) that foresee the development of guidance documents and IT-tools for the European Chemicals Agency, for industry and the authorities of the Member States including 5 central areas:

RIP 1 - REACH Process description

RIP 2- REACH-IT: Development of the IT system to support the REACH implementation

RIP 3 - Guidance Documents: Development of guidance documents for industry

RIP 4 - Guidance Documents: Development of guidance documents for authorities

RIP 5/6 - Setting up the Agency.

Here detailed information, documents and manuals on the REACH regulation are provided.

http://ec.europa.eu/enterprise/reach/index_en.htm

General information and several documents about REACH and GHS (Globally Harmonised System of Classification and Labelling of Chemicals) are available on the website of European Commission, Directorate General for Enterprise and Industry.

http://www.hse.gov.uk/reach/index.htm

General information are available on the website made by Health and Safety Executive (brief overview of REACH, pre-registration, case studies, etc.).

http://www.env-health.org/a/3022

General brief information about REACH are available on the website made by Health Environment Alliance (brief overview of REACH, pre-registration, FAQ about REACH, etc.).

http://www.rohs-international.com

A suite of simplified guidance notes for REACH is available on the website and a lot of services are provided in order to help above all that companies outside the EU will need to comply with REACH. This website is made by RoHS International.

http://www.iom-world.org/consulting/reach.php

IOM (Institute of Occupational Medicine) Consulting offers several services and information to get firms to find a way through these complex regulations. Few examples of the activities offered are providing guidance on how to ensure compliance that is specifically tailored for SMEs as well as providing services to major suppliers or users of chemicals, helping to identify and characterise relevant exposure scenarios, undertaking exposure modelling and/or measurement to inform the risk assessment process, advising on data gaps and helping firms fill them.