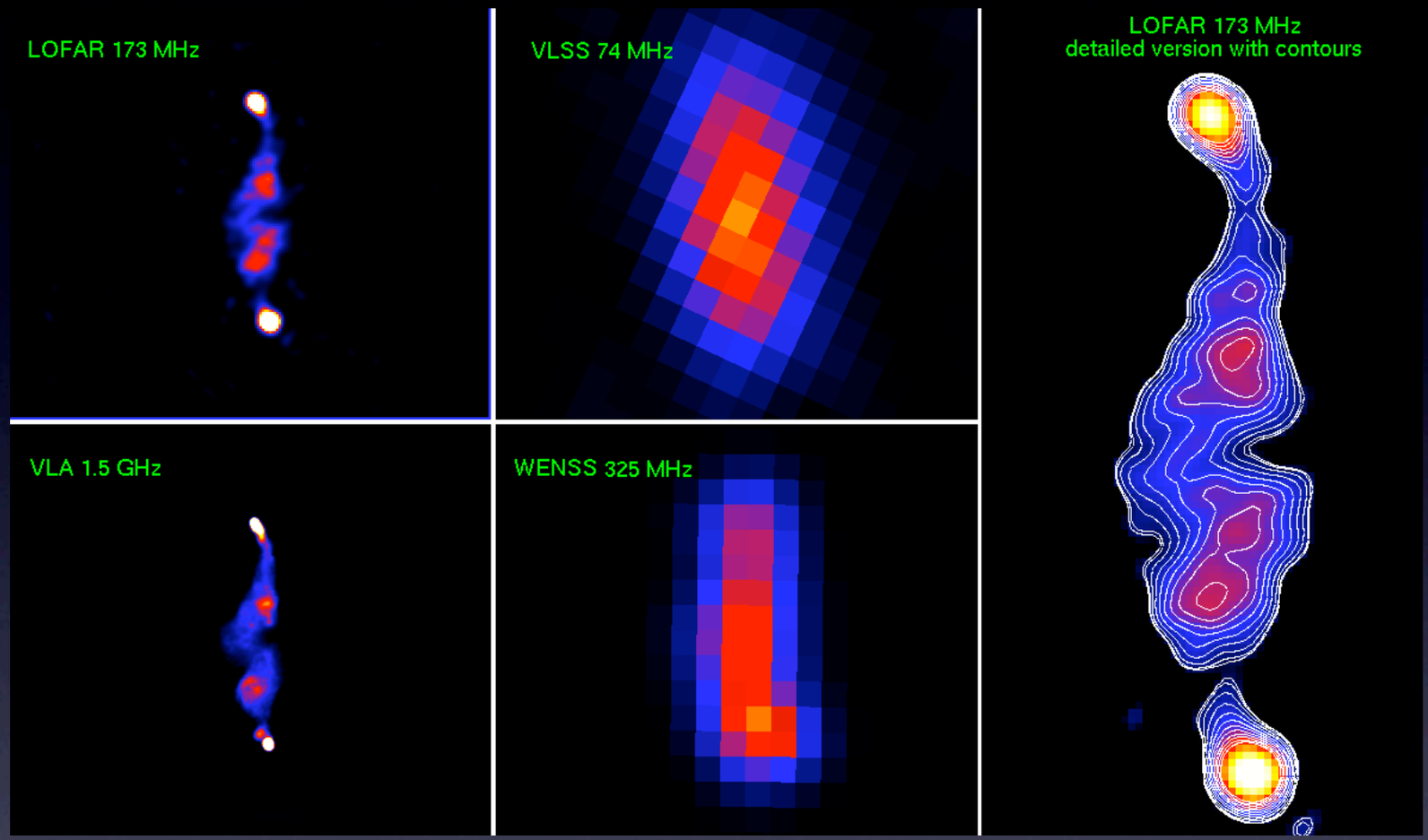
Getting started with LOFAR imaging



John Swinbank swinbank@transientskp.org

LOFAR Imaging Cookbook

Essential Reference!

Also check the wiki:
http://www.lofar.org/operations

The LOFAR Imaging Cookbook: Manual data reduction with the imaging pipeline

Written by Timothy Garn (and updated by Roberto Francesco Pizzo*)
with contributions from the LOFAR commissioning teams

Version 2.2.1: April 12, 2010

Abstract

This cookbook describes the process of manually reducing a measurement set with the LOFAR imaging pipeline. It is intended to speed up the learning process for future commissioning, by collating various tips, tricks and solutions in a single place. The LOFAR wiki[1] contains much more information on each stage of data reduction, but is out of date in many places. The LOFAR forum[2] should also be helpful for commissioning. The contents of this cookbook are at best an approximation to the correct way of reducing data, and at worst completely wrong – use with caution.

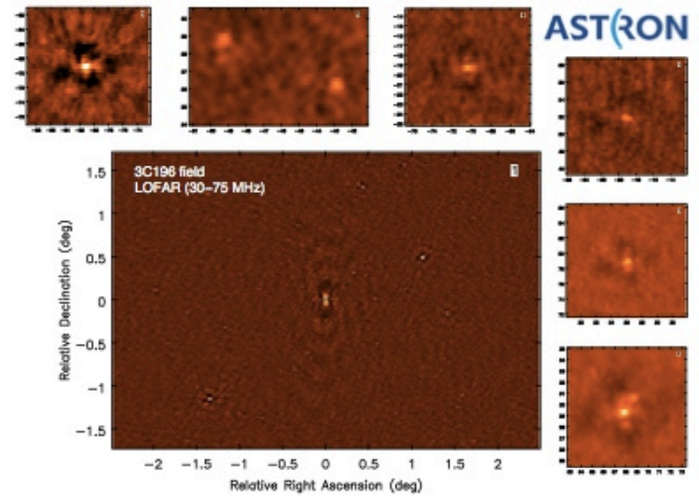
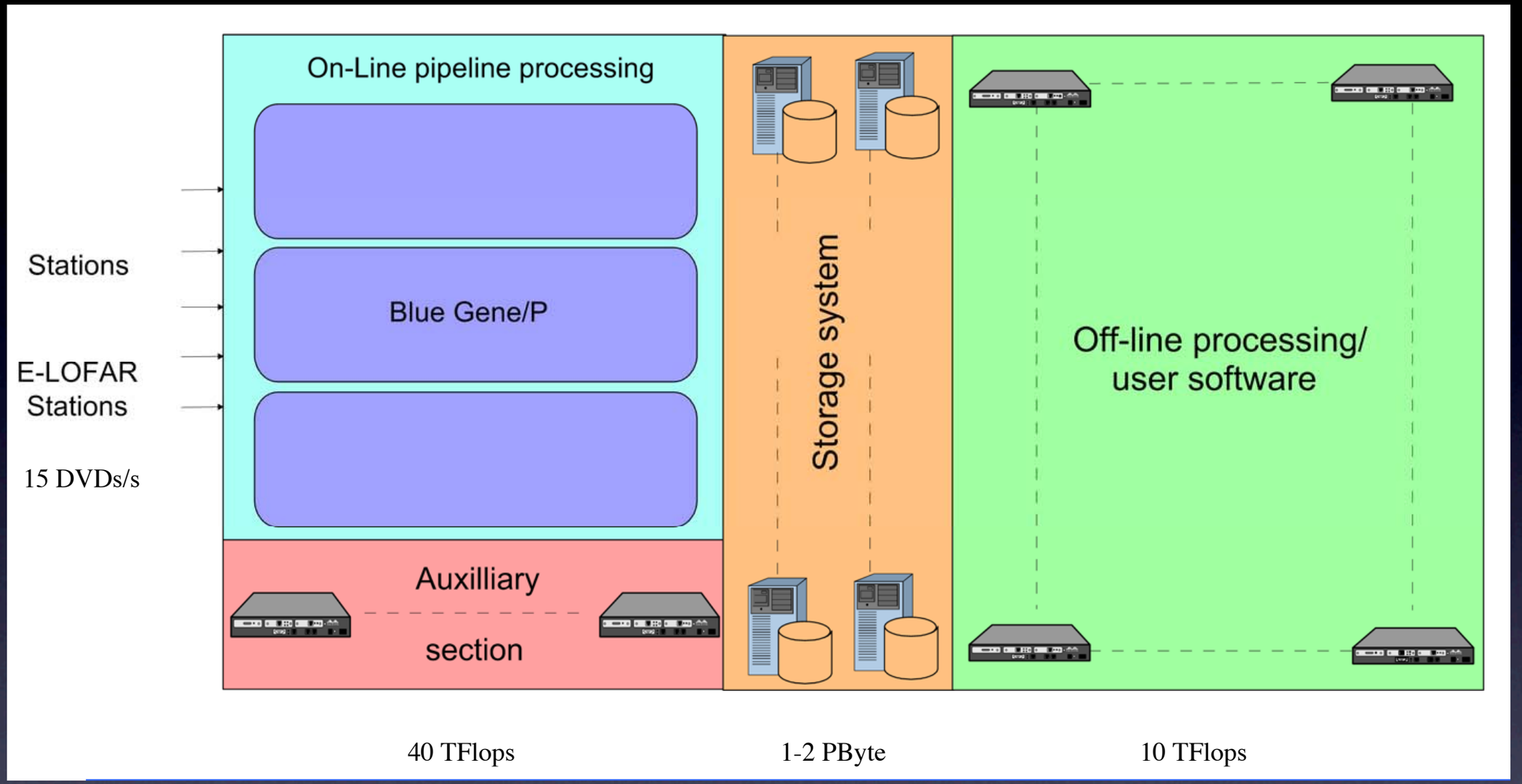
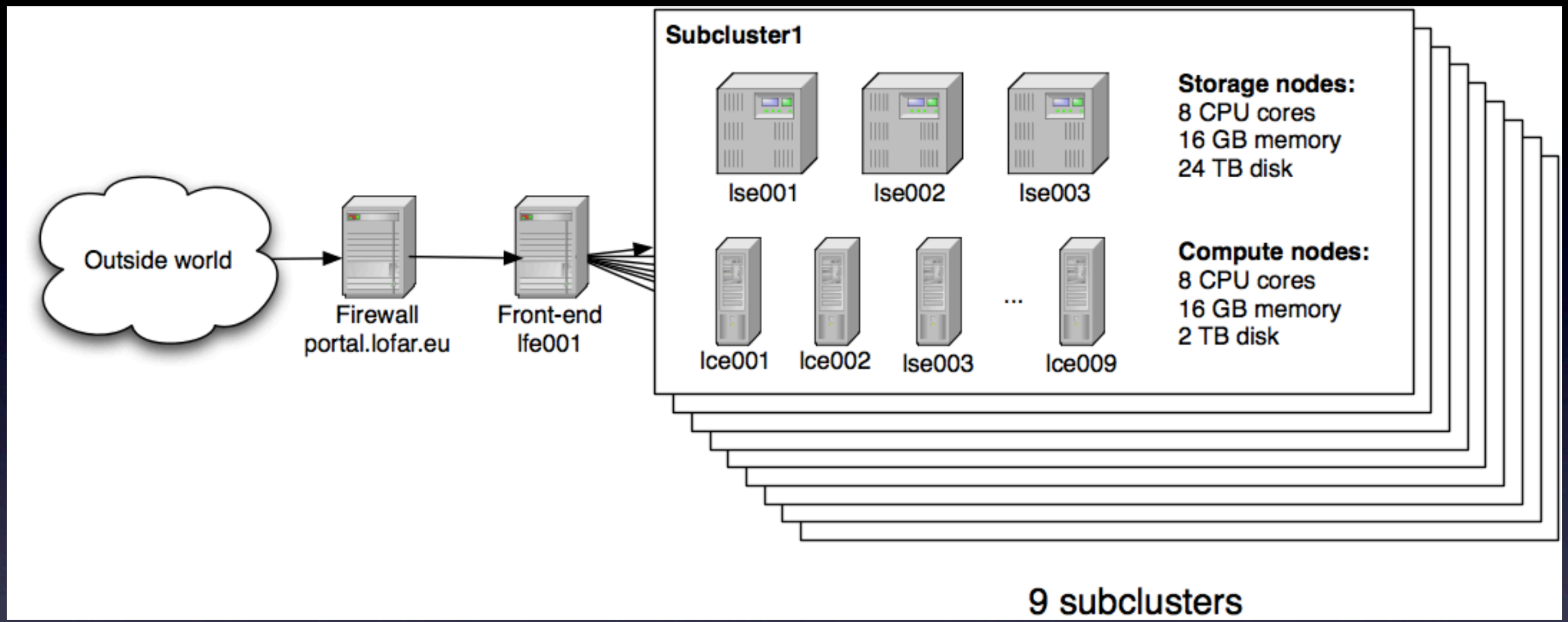


Figure 1: You too can make images like this with LOFAR



Data processing systems



Cluster layout

http://www.lofar.org/operations/doku.php?id=public:lofar_cluster

```
[swinbank@lfe001 ~]$ showsub
This script shows the subcluster definitions

sub   lce-nodes      lse-nodes      cexec-lce  cexec-lse  group     contact
====  ============   ============   =========  =========  ========  ======
sub1  lce001-lce009  lse001-lse003  lce:0-8    lse:0-2    product.  observers
sub2  lce010-lce018  lse004-lse006  lce:9-17   lse:3-5    TBB       ter Veen
sub3  lce019-lce027  lse007-lse009  lce:18-26  lse:6-8    imaging   Swinbank
sub4  lce028-lce036  lse010-lse012  lce:27-35  lse:9-11   pol/EOR   de Bruyn
sub5  lce037-lce045  lse013-lse015  lce:36-44  lse:12-14  pulsar    Hessels
sub6  lce046-lce054  lse016-lse018  lce:45-53  lse:15-17  busyweek5 Heald
sub7  lce055-lce063  lse019-lse021  lce:54-62  lse:18-20  develop.  Romein
sub8  lce064-lce072  lse022-lse024  lce:63-71  lse:21-23  busyweek5 Heald

You can use "cexec sub<n>: uptime", "cexec lse: uptime" or "cexec lce: uptime"
```

You will only be able to access certain subclusters

Disk layout

- Home directories shared across cluster
  - Only to subclusters where owner has access
- Storage nodes have 4 data partitions:
  - /data1 to /data4 on lseXXX
  - /net/subN/lseXXX/dataN on lceXXX
  - Only available within their subcluster
- /data/scratch on compute nodes

Clusterdesc files

- Machine readable description of subcluster layout
- Required by various imaging pipeline tools
- Most recent versions in ~diepen/cdesc

```
[swinbank@lfe001 ~]$ more /opt/lofar/etc/cdesc/
sub3.clusterdesc
ClusterName = sub3
NNodes = 12

# Storage nodes.
Node0.NodeName = lse007
Node0.NodeFileSys = [ lse007:/data1..4 ]
Node0.NodeMountPoints = [ /data1..4 ]
Node1.NodeName = lse008
Node1.NodeFileSys = [ lse008:/data1.. 4 ]
Node1.NodeMountPoints = [ /data1..4 ]
Node2.NodeName = lse009
Node2.NodeFileSys = [ lse009:/data1..4 ]
Node2.NodeMountPoints = [ /data1..4 ]

# Compute nodes.
Node3.NodeName = lce019
Node3.NodeFileSys = [lce019:/data, lse007..9:/data1..4 ]
Node3.NodeMountPoints = [/data, /net/sub1/lse007..9/data1..4 ]
Node4.NodeName = lce020
Node4.NodeFileSys = [lce020:/data, lse007..9:/data1..4 ]
Node4.NodeMountPoints = [/data, /net/sub1/lse007..9/data1..4 ]
[etc]
```

Login environment

- **You will need to set your environment up for whatever tools you want to run!**
- You can do this manually (`$PATH`, `$LD_LIBRARY_PATH`, `$PYTHONPATH`, etc)
- Or use the "LOFAR Login Environment"
  - http://www.lofar.org/wiki/doku.php?id=public:lle
  - "use PackageName"
  - Essential: "use LofIm"

A little more setup...

- Lots of tools depend on casacore data; tell them where to find it:

```
[swinbank@lfe001 ~]$ cat > ~/.casarc
measures.DE200.directory: /opt/casacore/data/ephemerides
measures.DE405.directory: /opt/casacore/data/ephemerides
measures.line.directory: /opt/casacore/data/ephemerides
measures.sources.directory: /opt/casacore/data/ephemerides
measures.comet.directory: /opt/casacore/data/ephemerides
measures.ierseop97.directory: /opt/casacore/data/geodetic
measures.ierspredict.directory: /opt/casacore/data/geodetic
measures.tai_utc.directory: /opt/casacore/data/geodetic
measures.igrf.directory: /opt/casacore/data/geodetic
measures.observatory.directory: /opt/casacore/data/geodetic
```

Parset files

- Configuration for many of the key tools.
- Simple list of key/value pairs; hierarchical structure via “.”, comments via “#”.

- Ask for examples!

```
msin.startchan = 8
msin.nchan = 240
msin.datacolumn = DATA
msout.datacolumn = DATA
steps = [flag1,flag2,avg1,flag3]
# Squashing pass to average all channels into one
avg1.type = squash
avg1.freqstep = 240
avg1.timestep = 1
[etc]
```

SSH keys

- Save time typing all those passwords
- Make it possible to distribute tasks

```
[swinbank@lfe001 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/swinbank/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/swinbank/.ssh/id_rsa.
Your public key has been saved in /home/swinbank/.ssh/id_rsa.pub.
The key fingerprint is:
93:a3:25:05:48:1a:32:1d:d7:97:8e:65:65:9d:d8:b0 swinbank@lfe001
[swinbank@lfe001 ~]$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
[swinbank@lfe001 ~]$ ssh lce019 # Look: no password!
[swinbank@lce019 ~]$
```

Input data

- Correlated data written to /data directories on storage nodes
  - Multiple storage nodes per dataset
    - One MeasurementSet for each subband (ie, range of frequences)
    - Raw data **cannot be understood by standard tools**
      - Process through `NDPPP` first

VDS files

- Describe:
  - Contents of MS
  - Layout of data on cluster
- Required by various imaging pipeline tools
- One VDS file to one MeasurementSet
- Combine to describe complete dataset

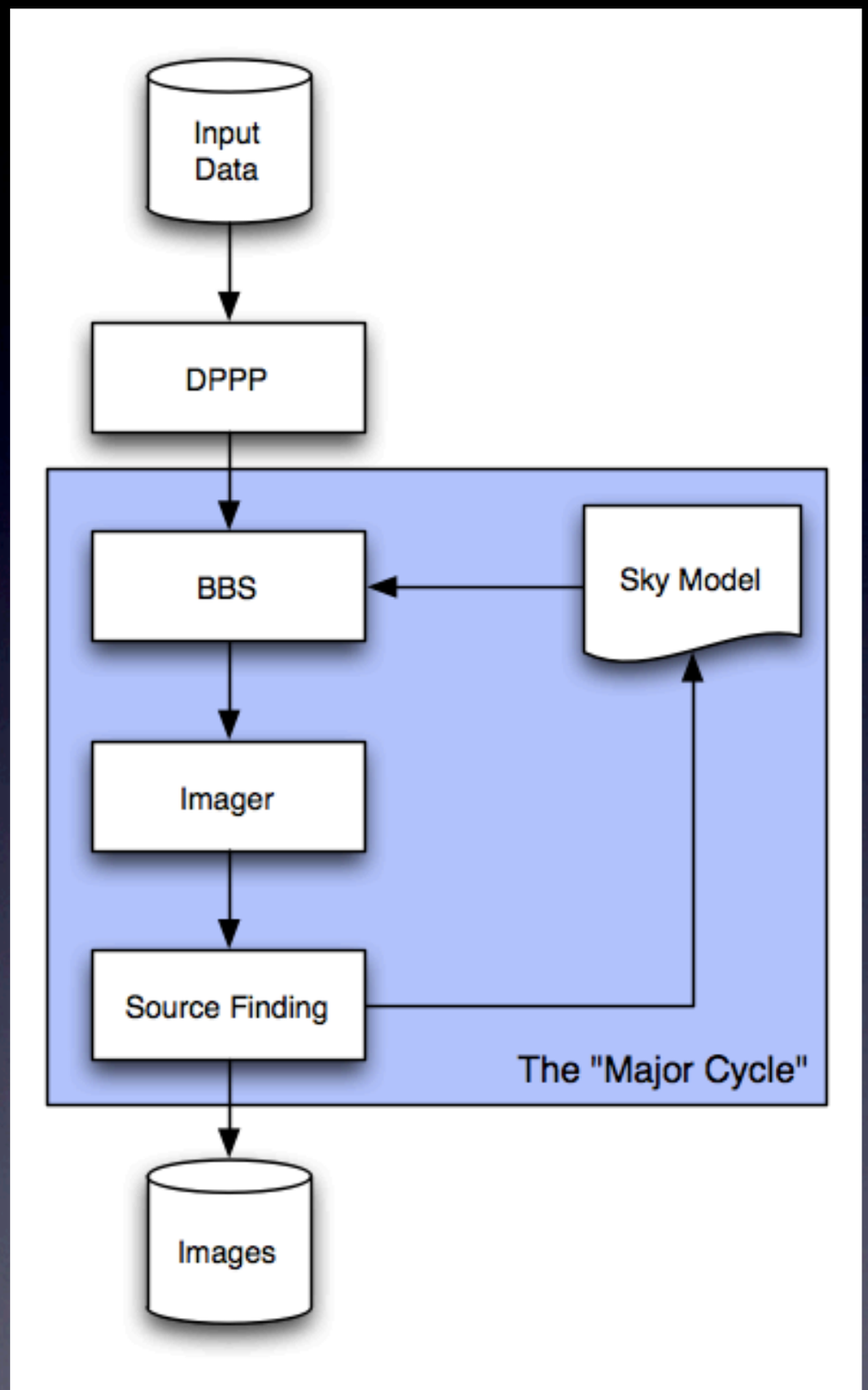
```
[swinbank@lfe001 ~]$ /opt/LofIm/daily/lofar/bin/makevds
Debug: registered context Global=0
Run as:  makevds clusterdesc ms [msvds] [hostname] [writetimes]
  default vds name is <ms>.vds
  default host name is gethostname()
  default writetimes is false (0)
[swinbank@lfe001 ~]$ /opt/LofIm/daily/lofar/bin/combinevds
Debug: registered context Global=0
Run as:  combinevds outName in1 in2 ...
```

Inspecting data

- `~rol/sw/bin/msinfo` provides information on a given MeasurementSet
- `~pizzo/EXAMPLES/Scripts` contains useful scripts
- CASA
  - Comprehensive NRAO package
- casacore
  - The “heart” of CASA
  - pyrap: Python interface (example later...)

The Standard Imaging Pipeline

We will consider each of the key components in more detail



DPPP

- Default Pre-Processing Pipeline
- Flagging & averaging of the data
- One subband (specified in parset) at a time
- **NB: prefer** `NDPPP` **to** `IDPPP`

BBS

- BlackBoard Selfcal
- Set up a personal BBS database: see wiki/cookbook for walkthrough
- Configuration via “sky model” and parset
  - Replicate the source parameters defined in the model in the output

Parallel BBS

- BBS can (...is designed to) run on many subbands at once, parallelized across multiple compute nodes
  - "calibrate" script makes this "easy"!
  - You need SSH keys set up (as discussed earlier), and clusterdesc & VDS files describing your data and the cluster

Bonus: extra flagging

- You can now image your calibrated data...
- ...and get something that looks like this. Great!

George Heald explains: BBS divides the DATA by the gains to get CORRECTED_DATA; very small gains will blow up the amplitudes in CORRECTED_DATA, which is probably what you see in your data. (This is normal.) And it explains why the images are garbage: all the signal that you see is based on bad solutions.

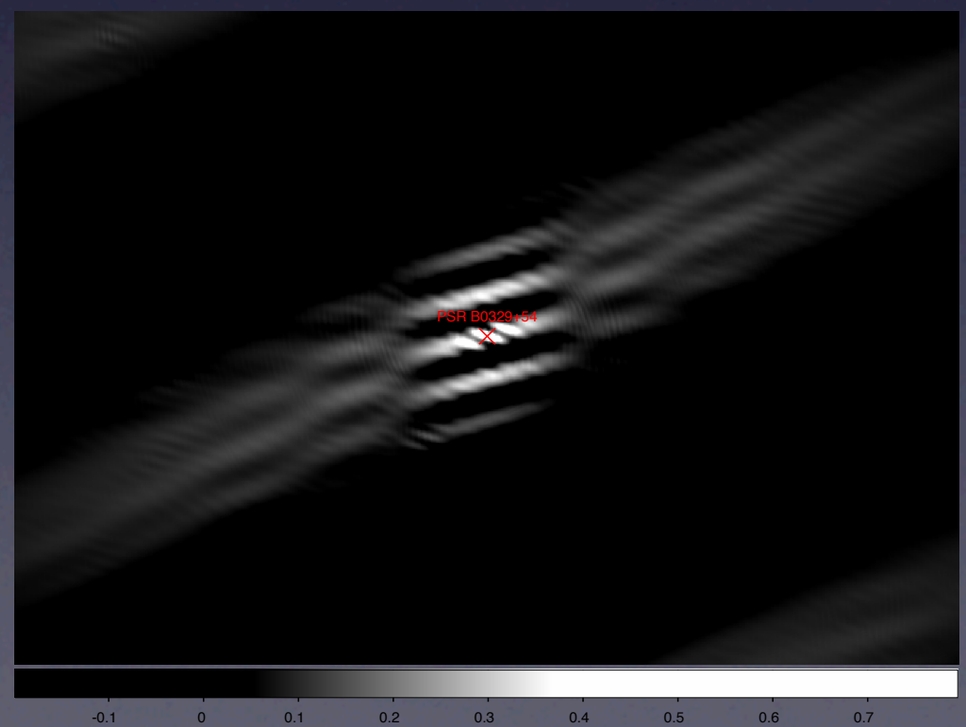
A quick fix

```
import sys, numpy
from pyrap.tables import table

def flag(input, max_value):
    t = table(input, readonly=False):
    for i, data in enumerate(t.getcol('CORRECTED_DATA')):
        if max([abs(val) for val in data[0]]) > max_value:
            t.putcell('FLAG', i, numpy.array([[True, True, True,True]]))
            t.putcell('FLAG_ROW', i, True)
    t.close()

if __name__ == '__main__':
    flag(sys.argv[1], float(sys.argv[2]))
```

Use pyrap to manipulate & experiment with the data



Imaging

- Three imagers:
  - CASApy (NRAO)
  - lwimager (casarest)
  - cimager (ASKAP)
- cimager is (likely) what will be deployed in standard LOFAR pipelines
- Others may be useful for experimentation

The Imaging Pipeline

- Integrated, automatic, managed by LOFAR control system, consistent logging, ...
- ...a whole extra layer of setup & configuration
- Concentrate on learning and understanding the individual tools first

Conclusions

- The only way to learn is to give it a try
- Be warned there are still many rough edges
- Keep records of everything that works!