Example of running full waveform source mechanism inversion using SeisSrcInv

This jupyter-notebook provides an example of how to use the python module SeisSrcInv to perform a full waveform source mechanism inversion. Firstly, an example of how to run an inversion is given using SeisSrcInv.inversion. The results of this inversion are then plotted using SeisSrcInv.plot.

In [1]:
```
# Import the module:
import SeisSrcInv
import matplotlib.pyplot as plt
```

1. Setup and perform a basic full waveform inversion

In [2]:
```
# Specify all inversion input variables:
datadir = 'data/real_and_greens_func_data'
outdir = 'data/FW_data_out'
real_data_fnames = ['real_data_ST01_z.txt', 'real_data_ST01_r.txt',
'real_data_ST01_t.txt', 'real_data_ST02_z.txt', 'real_data_ST02_r.t
xt', 'real_data_ST02_t.txt', 'real_data_ST03_z.txt', 'real_data_ST0
3_r.txt', 'real_data_ST03_t.txt'] # List of real waveform data file
s within datadir corresponding to each station (i.e. length is numb
er of stations to invert for)
MT_green_func_fnames = ['green_func_array_MT_ST01_z.txt', 'green_fu
nc_array_MT_ST01_r.txt', 'green_func_array_MT_ST01_t.txt', 'green_f
unc_array_MT_ST02_z.txt', 'green_func_array_MT_ST02_r.txt', 'green_
func_array_MT_ST02_t.txt', 'green_func_array_MT_ST03_z.txt', 'green
_func_array_MT_ST03_r.txt', 'green_func_array_MT_ST03_t.txt'] # Lis
t of Green's functions data files (generated using fk code) within
datadir corresponding to each station (i.e. length is number of sta
tions to invert for)
single_force_green_func_fnames = ['green_func_array_single_force_ST
01_z.txt', 'green_func_array_single_force_ST01_r.txt', 'green_func_
array_single_force_ST01_t.txt', 'green_func_array_single_force_ST02
_z.txt', 'green_func_array_single_force_ST02_r.txt', 'green_func_ar
ray_single_force_ST02_t.txt', 'green_func_array_single_force_ST03_z
.txt', 'green_func_array_single_force_ST03_r.txt', 'green_func_arra
y_single_force_ST03_t.txt'] # List of Green's functions data files
(generated using fk code) within datadir corresponding to each stat
ion (i.e. length is number of stations to invert for)
data_labels = ["ST01, Z", "ST01, R", "ST01, T", "ST02, Z", "ST02, R
", "ST02, T", "ST03, Z", "ST03, R", "ST03, T"] # Format of these la
bels must be of the form "station_name, comp" with the comma
inversion_type = 'DC' # Inversion type automatically filled (if sin
```

```
gle force, greens functions must be 3 components rather than 6) (Op
tions are: "DC", "single_force", "full_mt", "full_mt_Lune_samp", "D
C_single_force_couple", "DC_single_force_no_coupling", "DC_crack_co
uple", "single_force_crack_no_coupling"]: #["full_mt", "DC", "singl
e_force", "DC_single_force_couple", "DC_single_force_no_coupling",
"DC_crack_couple", "single_force_crack_no_coupling")
perform_normallised_waveform_inversion = False
compare_all_waveforms_simultaneously = False
num_samples = 1000 # Number of samples to perform Monte Carlo over
(typically might want to use 10e6, but used 1000 here for speed)
comparison_metric = "VR"
manual_indices_time_shift_MT = [9, -10, -9, 6, -15, -15, 8, 14, -13
]
manual_indices_time_shift_SF = [9, -11, -10, 6, -16, -16, 7, 13, -1
4]
cut_phase_start_vals = [0, 600, 600, 0, 575, 575, 0, 650, 650]
cut_phase_length = 150
nlloc_hyp_filename = "data/NLLoc_data/loc.Tom__RunNLLoc000.20090121
.042009.grid0.loc.hyp"
num_processors = 1 # Number of processors to run for (default is 1)
set_pre_time_shift_values_to_zero_switch = False # If True, sets va
lues before time shift to zero (default is True)
return_absolute_similarity_values_switch = True # If True, will als
o save absolute similarity values, as well as the normallised value
s.
# For other options, see help(SeisSrcInv.inversion.run())
```

In [3]:
```
# And perform inversion:
SeisSrcInv.inversion.run(datadir, outdir, real_data_fnames, MT_gree
n_func_fnames, single_force_green_func_fnames, data_labels, inversi
on_type, perform_normallised_waveform_inversion, compare_all_wavefo
rms_simultaneously, num_samples, comparison_metric, manual_indices_
time_shift_MT, manual_indices_time_shift_SF, nlloc_hyp_filename, nu
m_processors=num_processors, set_pre_time_shift_values_to_zero_swit
ch=set_pre_time_shift_values_to_zero_switch, return_absolute_simila
rity_values_switch=return_absolute_similarity_values_switch, cut_ph
ase_start_vals=cut_phase_start_vals, cut_phase_length=cut_phase_len
gth)
```

```
Saving FW inversion to file: data/FW_data_out/least_squares_result
/20090121042009185230_FW_DC.pkl
Saving FW inversion to file: data/FW_data_out/least_squares_result
/20090121042009185230_FW_DC.wfs

/Users/tomhudson/anaconda2/lib/python2.7/site-packages/SeisSrcInv-
0.0.2-py2.7.egg/SeisSrcInv/inversion.py:245: FutureWarning: `rcond
` parameter will change to the default of machine precision times
``max(M, N)`` where M and N are the input matrix dimensions.
To use the future default and silence this warning we advise to pa
ss `rcond=None`, to keep using the old, explicitly pass `rcond=-1`
.

Processing for process: 0 for  1000 samples.
Processor number: 0 - Processed for 0 samples out of 1000 samples
Finished processing process: 0 for  1000 samples.
Saving FW inversion to file: data/FW_data_out/20090121042009185230
_FW_DC.pkl
Saving FW inversion to file: data/FW_data_out/20090121042009185230
_FW_DC.wfs
Finished
```

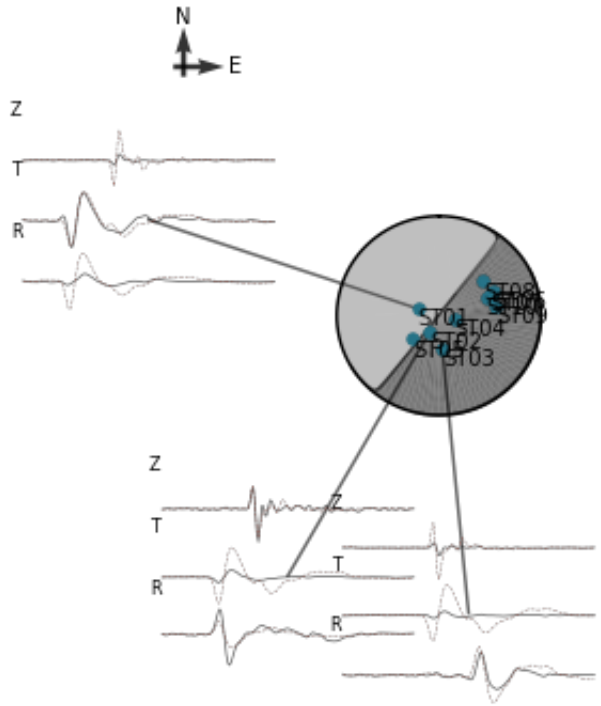
2. And plot results:

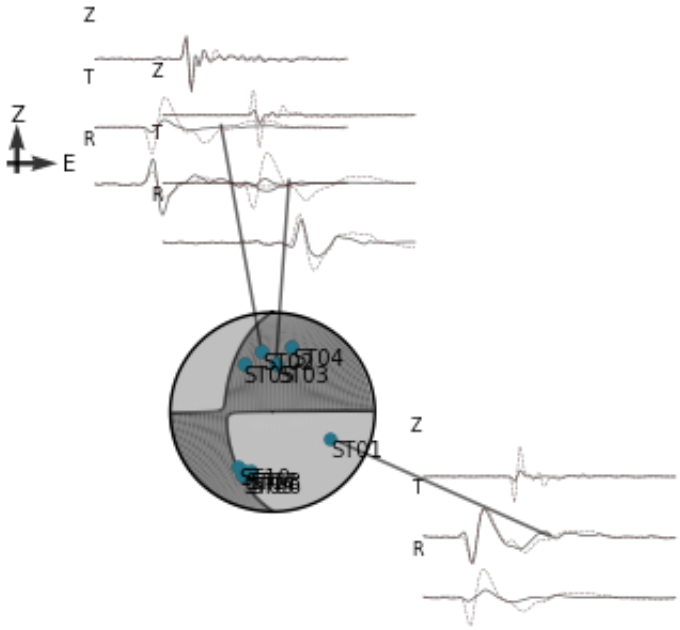
Note: Result may not be realistic since only performed for a small number of samples above.

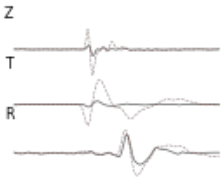
In [4]:
```
# Specify plotting parameters:
inversion_type = "DC" # Must be same as above
inversion_output_data_dir = "data/FW_data_out"
event_uid = "20090121042009185230" # UID, as saved for output files
plot_wfs_on_focal_mech_switch= True # If True, plots waveforms on f
ocal mechanism plot (default is True)
# For other options, see help(SeisSrcInv.plot.run())
```

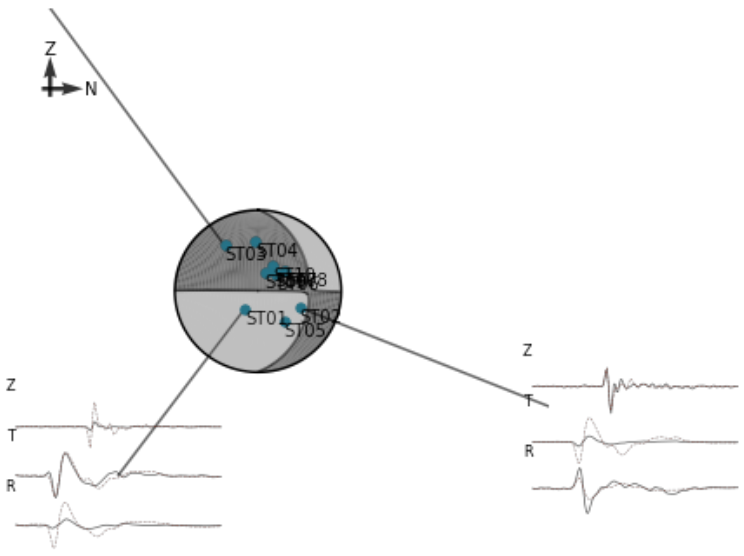
In [5]:
```
# Run the plotting:
SeisSrcInv.plot.run(inversion_type, event_uid, inversion_output_dat
a_dir,plot_wfs_on_focal_mech_switch=plot_wfs_on_focal_mech_switch)
# (Note: Saves output to files in directory "Plots")
```

```
Plotting data for inversion
Processing data for: data/FW_data_out/20090121042009185230_FW_DC.p
kl
Full MT (max prob.):
[[  -4.95871957    6.38709517  268.21682918]
 [   6.38709517   -8.22691908 -345.4775751 ]
 [ 268.21682918 -345.4775751    13.18563866]]
(For plotting radiation pattern)
Finished processing unconstrained inversion data for: data/FW_data
_out/20090121042009185230_FW_DC.pkl
Finished
```




Z
T
Z
Z
R
T
E
R
ST04
ST05
ST03
ST01
Z
T
R


Z
T
R


Z
N
ST03
ST04
ST01
ST05
Z
T
R
Z
T
R

In [ ]: